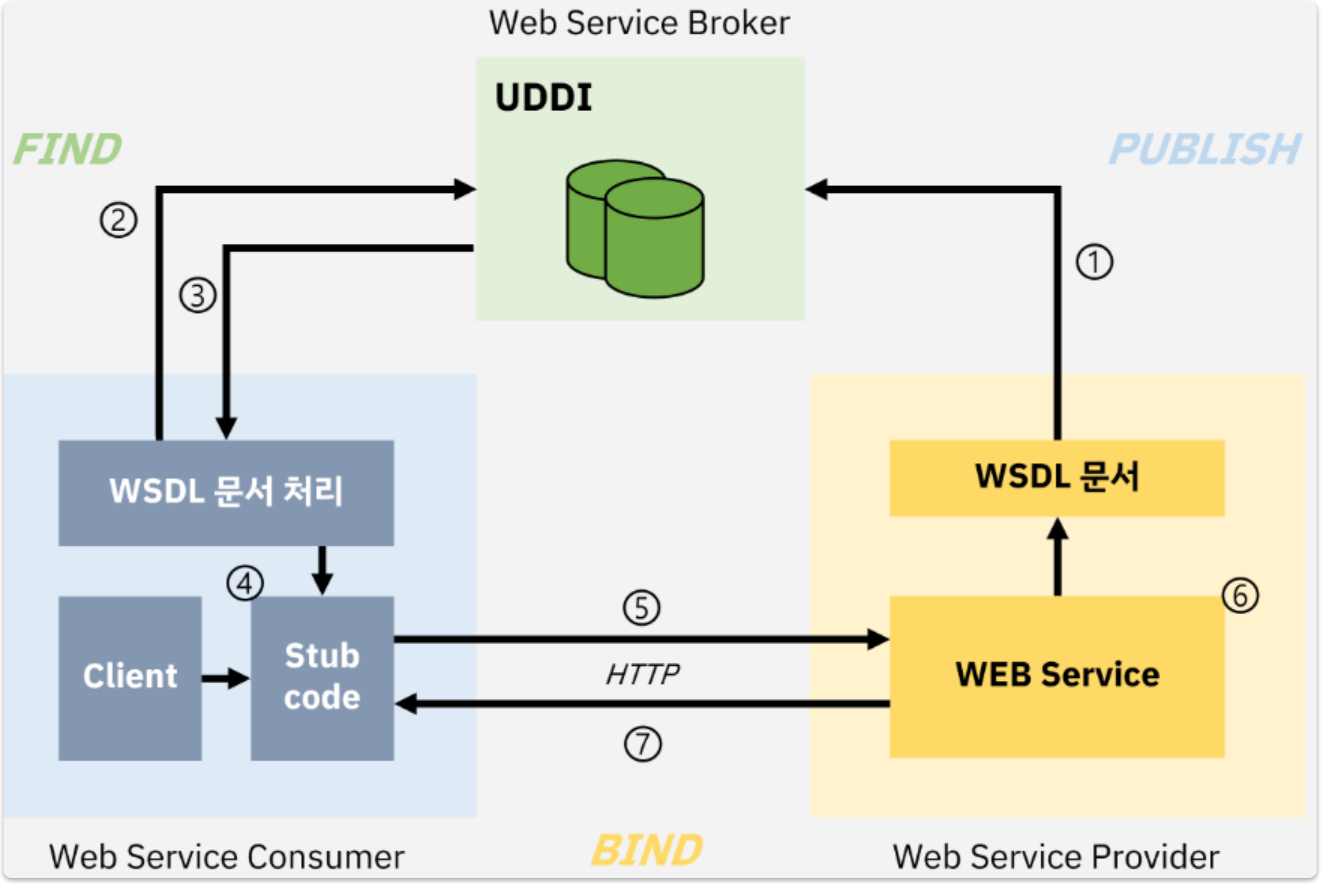
SOAP

#SOAP #UDDI #WSDL
URL : https://gruuuuu.github.io/programming/soap/

01. SOAP란?

> *Simple Object Access Protocol*

- HTTP, HTTPS, SMTP 등을 사용하여 XML 기반의 메시지를 컴퓨터 네트워크 상에서 교환하는 통신 프로토콜
- 서로 다른 Service들간의 연동을 목적으로 상호 이해 가능한 포맷의 메세지를 송/수신함으로써 원격지에 있는 서비스 객체나 API를 자유롭게 사용가능한 프로토콜

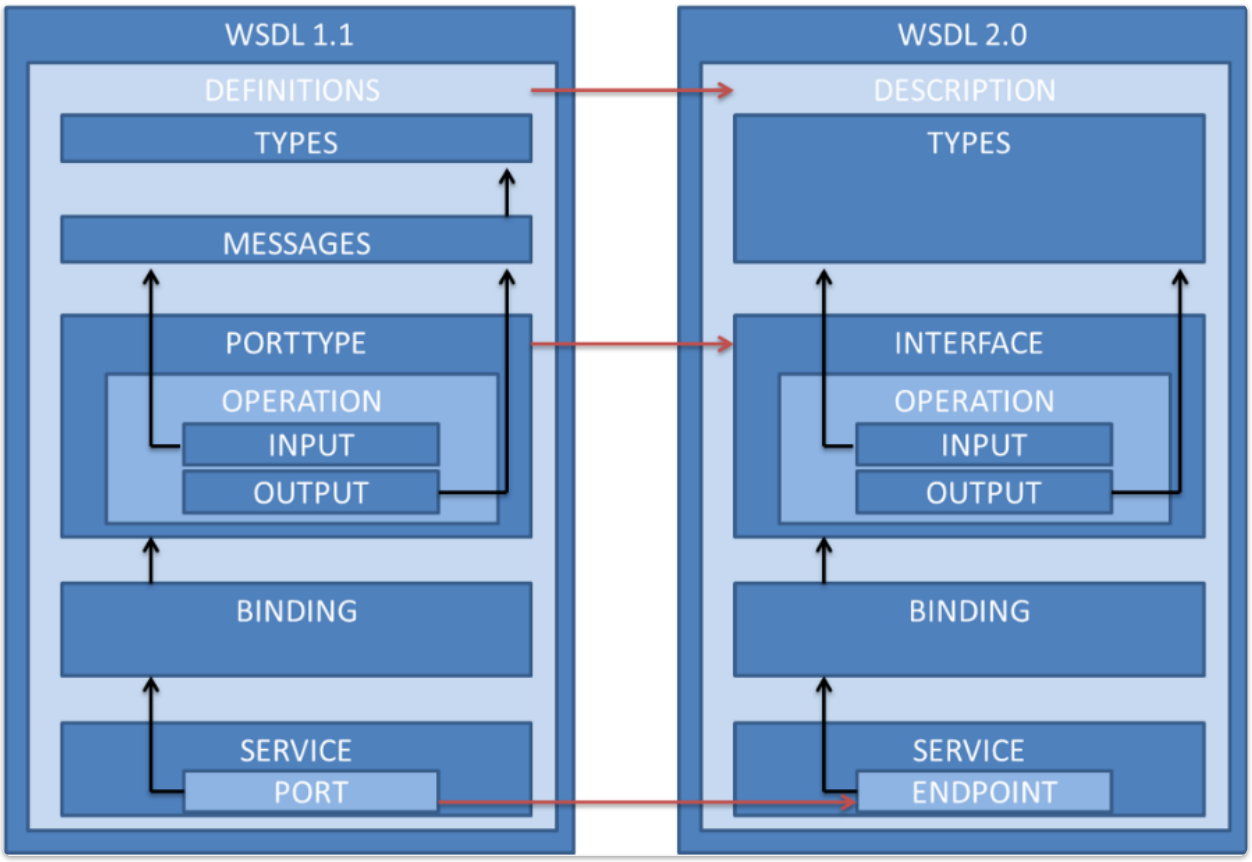


---

02. WSDL란?

> *Web Services Description Language*

- 웹 서비스가 기술된 정의 파일의 총칭으로 XML을 사용해 기술됩니다.
- 웹서비스의 구체적인 내용이 기술되어있는 문서이고 서비스 제공 장소, 서비스 메세지 포맷, 프로토콜들이 기술되어 있습니다.



- Types : 교환될 메세지 설명, 사용될 데이터 형식 정의
- Interface : Operation 정의 (Input & Output)
- Binding : Interface에 정의된 작업에 대해 메세지 형식과 프로토콜 정의 (Class화)
- Service : WebService URL endpoint 정의

02.01. Type

> *교환할 메세지를 설명하고, 사용될 데이터 형식을 정의하게 됩니다.*

```
<wsdl:types>
...
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:tns="http://example.org/calculator"
               targetNamespace="http://example.org/calculator"
               elementFormDefault="qualified"
               attributeFormDefault="qualified">
        <xs:element name="AddValuesRequest" type="tns:AddValuesType" />
        <xs:element name="AddValuesResponse" type="tns:AddValuesResponseType" />

        <xs:complexType name="AddValuesType">
            <xs:sequence>
                <xs:element name="FirstValue" type="xs:int" minOccurs="1" maxOccurs="1" />
                <xs:element name="SecondValue" type="xs:int" minOccurs="1" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>

        <xs:complexType name="AddValuesResponseType">
```

```
            <xs:sequence minOccurs="1" maxOccurs="1">
                <xs:element name="Result" type="xs:int" />
            </xs:sequence>
        </xs:complexType>
...
    </xs:schema>
</wsdl:types>
```

- Schema를 통해 필요한 부분을 import하고 wsdl에서 사용할 데이터 형식을 정의합니다.
- *AddValuesRequest*와 *AddValuesResponse*을 선언하고 데이터 형식을 *<xs:complexType>* 안에 정의하고 있습니다.

02.02. Interface

> *Operation 즉, 함수를 정의하게 됩니다.*

```
<wsdl:interface name="CalculatorInterface">
    <wsdl:fault name="fault" element="calc:CalculationFault" />
    <wsdl:operation name="AddValues" pattern="http://www.w3.org/ns/wsdl/in-out"
style="http://www.w3.org/ns/wsdl/style/iri" wsdl:safe="true">
        <wsdl:documentation>Adds up passed values and returns the result</wsdl:documentation>
        <wsdl:input messageLabel="in" element="calc:AddValuesRequest" />
        <wsdl:output messageLabel="out" element="calc:AddValuesResponse" />
        <wsdl:outfault messageLabel="fault" ref="tns:fault" />
    </wsdl:operation>
</wsdl:interface>
```

- Interface Name : CalculatorInterface
- Fail : *type*에서 정의했던 *CalculationFault* Type 메세지 출력
- Operation Name : AddValues
    - Input Type : *type*에서 정의했던 *AddValuesRequest*
    - Output Type : *type*에서 정의했던 *AddValuesResponse*
- Input과 Output의 순서도 중요! 총 4가지가 있다.
    - One-Way : Only Input (Client to Server)
    - Request-Response : Client =Req⟹ Server & Server =Res⟹ Client

02.03. Binding

> *Interface에 정의된 작업에 대해 메세지 형식과 프로토콜을 정의하게 됩니다.*

```
<wsdl:binding name="CalculatorBinding" interface="tns:CalculatorInterface"
type="http://www.w3.org/ns/wsdl/soap"
soap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">
    <wsdl:operation ref="tns:AddValues" soap:mep="http://www.w3.org/2003/05/soap/mep/soap-
response" />
```

```
    <wsdl:fault ref="tns:fault" soap:code="soap:Sender" />
</wsdl:binding>
```

- *CalculatorInterface* Interface에 대해 Prococol로 `#soap` 을 사용

02.04. Service

> *WebService URL endpoint 정의*

```
<wsdl:service name="CalculatorService" interface="tns:CalculatorInterface">
    <wsdl:endpoint name="CalculatorEndpoint" binding="tns:CalculatorBinding"
address="http://localhost:8080/services/calculator" />
</wsdl:service>
```

- *CalculatorService*에 대한 요청은 *CalculatorEndpoint*에서 처리

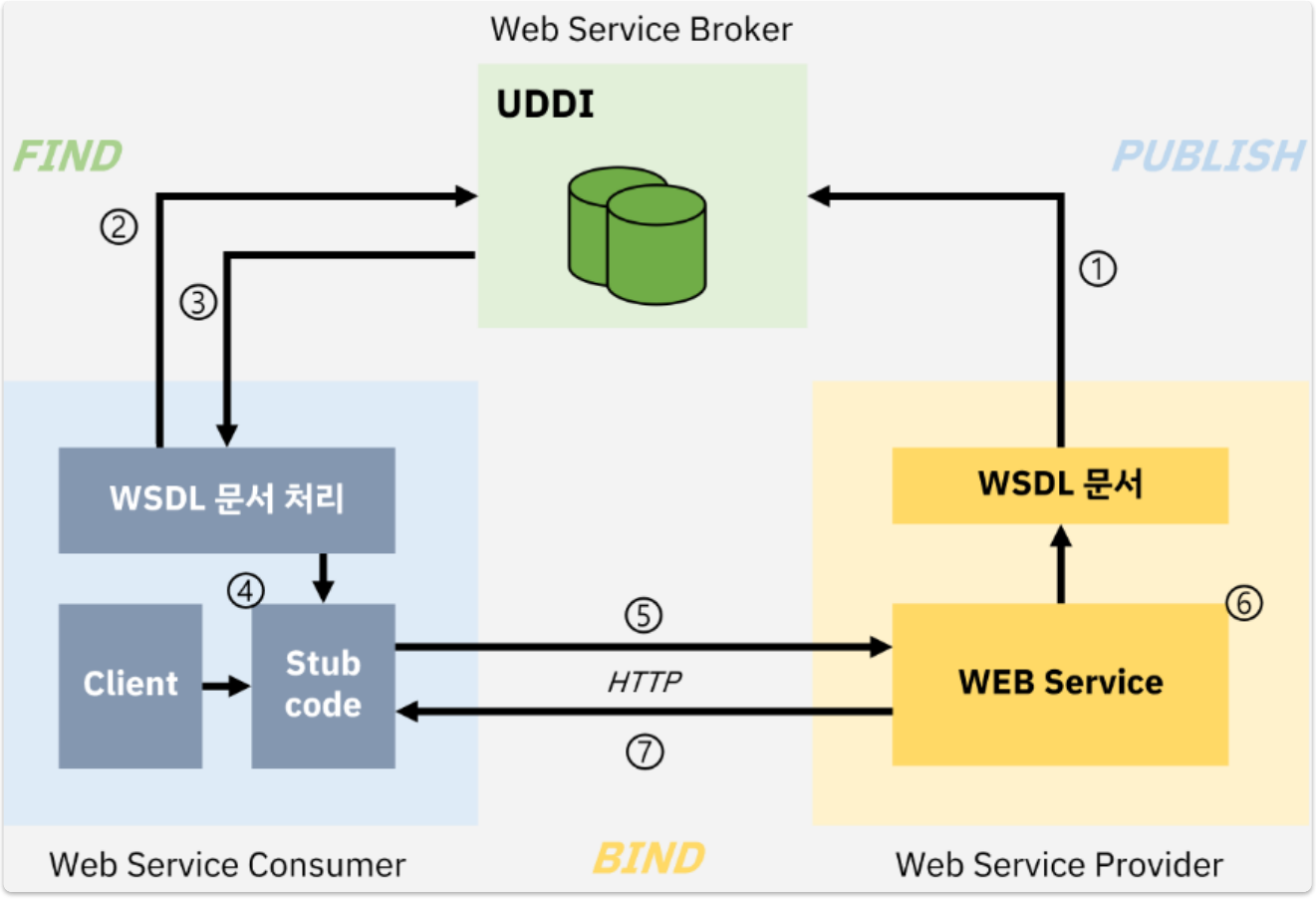
---

03. UDDI란?

> *Universal Description, Discovery and Integration*

- public registry
  - 전역 저장소이기 때문에 공개적으로 접근하고 WSDL을 검색할 수 있습니다.
- WSDL을 담아둡니다.

04. 동작 방식



- Web Service Broker : UDDI, 서비스 등록 및 검색, 저장, 관리하는 주체
- Web Service Provider : 웹 서비스를 구현하여 운영하고 제공하는 주체
- Web Service Consumer : 웹 서비스를 요청하는 주체
- Sequence
  1. 서비스 제공자는 UDDI에 사용가능한 WSDL 등록
  2. 서비스 사용자는 원하는 서비스를 위해 UDDI를 검색
  3. 원하는 서비스에 대한 WSDL Download
  4. WSDL 문서를 처리하여 적절한 인터페이스에 맞게 SOAP Message 작성
  5. HTTP를 통해 서비스 요청
  6. 서비스 제공자는 요청 값을 Decording → 적절한 서비스 로직 수행
  7. 처리 결과 SOAP MEssage로 작성 후 HTTP를 통해 요청자에게 반환

---

05. 특징

- XML기반으로 플랫폼에 독립적, 서로 다른 운영체제에서 실행되는 서비스간 통신 방법을 제공
- 프록시와 방화벽에 구애받지 않고, HTTP, HTTPS등을 통해 메세지 교환
- 확장 용이
- 에러 처리에 대한 내용이 기본적으로 내장
- XML형태로 메세지를 보내기 때문에 다른 기술들에 비해 상대적으로 처리속도가 느림