

Date _____
Page _____

* What is XML? Write its Features and Advantages. Create DTD File for student information and create valid well-formed XML document to store student information with DTD File.

=> XML :

XML stands for extensible Markup Language much like HTML.

XML is designed to store and transport data and to be self-descriptive.

XML language does not have any predefined tags.

XML can be save using .xml extension.

XML simplifies Data Sharing, Data Transport and Data availability.

Syntax :

<root >

<child >

<subchild > </subchild >

<child >

<root >

All the User defined tages are case sensitive.

→ XML Features :

- 1 XML Language is Human and computer friendly format.
- 2 XML has tree structure for Data Handle like root, child etc.
- 3 XML can handle complex structure data for a long time.
- 4 XML has a User defined structure to create any XML document.

5 In XML, Data is described using markup Language.

6 XML uses a Document Type Definition or XML Schema for describe the data.

7 XML is Platform independent.

→ XML Advantages

1 XML Document is plain text and human readable.

2 XML Document has a tree structure to handle complex data.

3 XML Files are Operating System independent.

4 XML Document is allows content reuse.

5 XML is allows users to defines their own tags.

6 XML is extendable.

→ XML File:

```
<?xml version="1.0"?>  
<!DOCTYPE Student-Info SYSTEM "  
C:/Users/Lenovo/Desktop">
```

```
<Student-Info>
```

```
<Student>
```

```
<name>Khushi Gandhi</name>
```

```
<id>21BEIT30026</id>
```

```
<age>19</age>
```

```
<address>Godhra</address>
```

```
</Student>
```

```
</Student-Info>
```

→ DTD File:

```
<?xml version="1.0"?>
```

```
<!ELEMENT Student-Info (Student*)>
```

```
<!ELEMENT Student-Info (name, id,  
age, address)>
```

```
<!ELEMENT name (#PCDATA)>
```

```
<!ELEMENT id (#PCDATA)>
```

<!ELEMENT age(<#PCDATA)>

<!ELEMENT address(<#PCDATA)>

* What is XML Schema. Why we need XML Schema. Create XML schema for student information and create a valid well-formed XML.

=> XML Schema:

XML Schema is used to define structure of an XML document.

XML Schema is also called XML Schema Definition.

XML Schema is used to validate the XML Document data.

XML Schema is defined whole XML document in the form of tree structure.

XML Schema alternative to DTD use.

XML Schema can be save using .xsd extension.

-> Why we need XML Schema:

The main purpose of an XML Schema is to define the legal building blocks of an XML Document.

XML Schema contains element and attributes that can appear in a document.

-> Student Information XML File:

```
<?xml version = "1.0"?>  
<Student_Inf  
  xmlns:xsi = "http://www.w3.org/2001  
  XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation  
  = "C:\Users\Lenovo\Desktop">
```

```
<student>
```

```
<name> Khushi Gandhi </name>  
<Id> 26 </Id>  
<age> 19 </age>
```

```
</student>
```

```
</Student_Inf>
```

-> XML Schema:

```
<?xml version="1.0"?>
```

```
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema">
```

```
<xs:element="StudentInfo"  
  type="xs:string">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element="Student"  
  type="xs:string">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element="name"  
  type="xs:string">
```

```
<xs:element="ID"  
  type="xs:decimal">
```

```
<xs:element="age"  
  type="xs:decimal">
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:schema>
```

* What is XSL in XML. Using XSL display student information in tabular format.

=> XSL :

XSL stands for Extensible Stylesheet Language.

XSL is used to style the XML document like CSS styling.

XSL file can be saved using .xsl extension.

XSL provides an easy way to merge XML data into the presentation.

-> Student information XML File :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
  href="C:\Users\Lenovo\Desktop"?>
<Student-Info>
```

```
<Student>
```

```
<name> Gandhi Khushi </name>
<Id> 26 </Id>
<age> 19 </age>
```

</Student>

<Student>

<name> Gandhi khush </name>

<Id> 2 </Id>

<age> 19 </age>

</Student>

</Student-Info>

-> Student Information XSL File:

<?xml version="1.0" encoding="UTF-8"?

<xsl:stylesheet version="1.0"?

xmlns:xsl="http://www.w3.org/1999/xsl/Transform"?

<xsl:template match="/"?

<html>

<body>

<h1 align="center">Student
Details </h1>

<table border="3" align="center":

<tr>

<th> Name </th>

<th> Id </th>

<th> Age </th>

</tr>

```
<xsl:for-each select = "Student-Info  
/Student">
```

```
<xsl:sort select = "age" />
```

```
<tr>
```

```
<td><xsl:value of select = "name">  
</td>
```

```
<td><xsl:value of select = "age"></td>
```

```
<td><xsl:value of select = "Id">  
</td>
```

```
</tr>
```

```
</xsl:for-each>
```

```
</table>
```

```
</body>
```

```
</html>
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

* What is SOA. Explain the Fundamental of SOA

=> SOA:

SOA is stands for Service-Oriented Architecture which defines a way to make

software components.

SOA is define way, how services communicate with each other and provides Architectures of Services.

SOA is define how Services Provider and Services consumer communicate with each other using interface.

SOA is Provides all the services specification in Services registry.

Using SOA, Service Provider can implements any services specification.

Using SOA, Service consumer can request to consume the services.

The SOA services are easily available to any service requester or consumer.

The SOA services is easier to to test and deployed any where.

Fundamental of SOA :

SOA allows users to combine a large number of facilities in a one services.

SOA Provides loosely couples of services.

SOA can reduced cost of application development and deployment.

SOA can Provides Autonomy in which whole services have control over the logic.

SOA Services can be easily available to anyone on request.

SOA applications are more reliable because it is easy to maintain large service.

Date _____
Page _____

* What is SOAP and explain soap message structure with its syntax and its elements.

=> SOAP stands for Simple Object Access Protocol which is use to exchange data between two nodes.

SOAP uses XML Formate to transfer messages and works on top of application layer protocol.

SOAP is a light weighted data interchange protocol.

SOAP is mainly used for Web Services and Application Programming Interfaces.

-> Structure of SOAP Message :

SOAP Message contains Four elements.

- a) Envelope
- b) Header
- c) Body
- d) Fault

a Envelope:

SOAP Envelope is the root element in every soap message.

Envelope element contains two child element which is header and body.

b Header:

SOAP Header is a child element of SOAP Envelope which is optional element in message.

SOAP Header contain application related information in SOAP message.

c Body:

SOAP Body is a child element of SOAP Envelope which is mandatory element in message.

SOAP Body contains actual SOAP message intended for the ultimate endpoint of the message.

d Fault:

SOAP Fault is a child element of SOAP body which is ^{no} appear more than once in body element.

SOAP Fault element contains error message in soap message.

Syntax:

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
```

```
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
```

```
<soap:Header>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
<soap:Fault>
```

```
</soap:Fault>
```

```
</soap:Body>
```

</soap: Envelope >

=> Explain SOAP XML Request and SOAP XML Response structure.

SOAP XML Request is used by Service requestor to use the service.

SOAP XML Response is provide by Service Provider to provide services.

Using XML Protocol, SOAP Request and SOAP Response is generate.

Example:

- SOAP XML Request:

```
POST / InStock HTTP/1.1  
Host: www.thebrainspot.in  
Content-Type: application/soap+xml;  
charset = utf-8  
Content-Length: nm
```

<?xml version = "1.0" ?>

<soap: Envelope

Date _____
Page _____

```
xmlns:soap = "http://www.w3.org/2003/05/soap-envelope/"  
soap:encodingStyle = "http://www.w3.org/2003/05/soap-encoding"
```

```
<soap:Body xmlns:m = "http://www.thebrainspot.in/aboutus" >
```

```
<m:GetName > Founder  
<m:Name > Founder </m:Name >  
</m:GetName >  
</soap:Body >
```

```
</soap:Envelope >
```

- SOAP XML Response:

```
HTTP/1.1 200 OK  
Content-Type: application/soap+xml  
charset = utf-8  
Content-Length: nnn
```

```
<?xml version = "1.0"? >
```

```
<soap:Envelope
```

```
xmlns:soap = "http://www.w3.org/2003/05/soap-envelope/"  
soap:encodingStyle = "http://www.w3.org/2003/05/soap-encoding"
```

```
<Soap:Body xmlns:m = "https://  
www.thebrainspot.in/about  
us" >
```

```
<m:GetNameResponse >  
  <m:Name > Khushi Gundhi  
  </m:Name >  
</m:GetNameResponse >
```

```
</Soap:Body >
```

```
</Soap:Envelope >
```

* Explain SOAP Fault with its structure.

=> SOAP Fault element is a child element of SOAP Body.

A SOAP Fault element can only appear once in a SOAP message.

There are four subelements in SOAP Fault element.

ci) Faultcode : This subelement contain a code for identify the code.

cii) FaultString : This subelement of Fault contains human understandable explanation of fault.

ciii) FaultFactor : The subelement contains the URI of the SOAP that generated the fault.

civ) detail : The subelement contains specific error information related to the body element.

Syntax :

<soap: Body >

 + <soap: Fault >

 <FaultCode > - - - - </Faultcode>

 <FaultString > - - - - </Faultstring>

 <FaultFactor > - - - </Faultfactor>

 <detail > - - </detail>

 </soap: Fault >

</soap: Body >

* Write the difference between SOAP and REST Web Service.

| => | SOAP | REST Web Service |
|----|--|---|
| 1 | Stands for Simple Object Access Protocol. | Stands for Representational State Transfer. |
| 2 | Requires more Bandwidth. | Requires Less Bandwidth. |
| 3 | Less Preferred compared to REST. | More Preferred compared to SOAP. |
| 4 | Use XML Format only. | Use XML, HTML, Plain text format. |
| 5 | SOAP can not use REST. | REST can use SOAP. |
| 6 | Works over HTTP, HTTPS, SMTP, XMPP Protocol. | Works over HTTP and HTTPS Protocol only. |
| 7 | Follow Strict structure. | Follow Less strict structure. |

* What is WSDL?

→ WSDL stands for Web Services Description Language which is used to describe web services.

WSDL document is written in XML and specifies the location and method of web services.

There are two parts of WSDL.

ci) Abstract

cii) Concrete

ci) Abstract: This part contains web service types, message, operation and port type.

cii) Concrete: This part contains web service message bindings, method, services and ports.

→ Explain WSDL structure of document.

WSDL Document contains four types of elements in document.

- a) types
- b) message
- c) port Type
- d) binding.

a types:

This element is defined all the data types which is used in the web services.

b message:

This element is defined all the data elements for each operation which is used in the web services.

c portType:

This element is defined all the operation that can be performed in the web services.

In PortType, In Operation element we can write operation.

d binding:

This element is defined all the

protocol and data format for each port type in web service.

Syntax:

<types>

</types>

<message>

</message>

<portType>

</portType>

<binding>

</binding>

* What is UDDI. Why we need UDDI?

=> UDDI stands for Universal Description, Discovery and Integration

UDDI is support developers to find the services.

UDDI communicates via SOAP and built into the .NET Platform.

UDDI uses WSDL to describe interfaces to web services.

UDDI defines a way to publish and discover information about web service.

UDDI is a SOAP-based protocol that defines how clients communicate with UDDI registries.

UDDI includes an XML schema for SOAP message that defines a set of document to describe services.

* Explain UDDI Registry with its components.

⇒ UDDI Registry is XML-based registry for provides web services description.

There are three components of UDDI Registry.

- ci) Yellow Page
- cii) White Page
- ciii) Green Page

ci) Yellow Page :

The Yellow Page contains all the business classifications.

Yellow Page contain more details about the services.

The Yellow Page classifies business according to the NAICS.

cii) White Page :

The white pages contain basic information of a business.

The white page contain company's basic information like name, address, contact phone number etc.

ciii) Green Page :

Green Pages contains technical information about a web service.

Green Page allows someone to bind to a web service after it is found.

Green Page includes,

- Various interfaces
- URL Locations
- Discovery information etc.

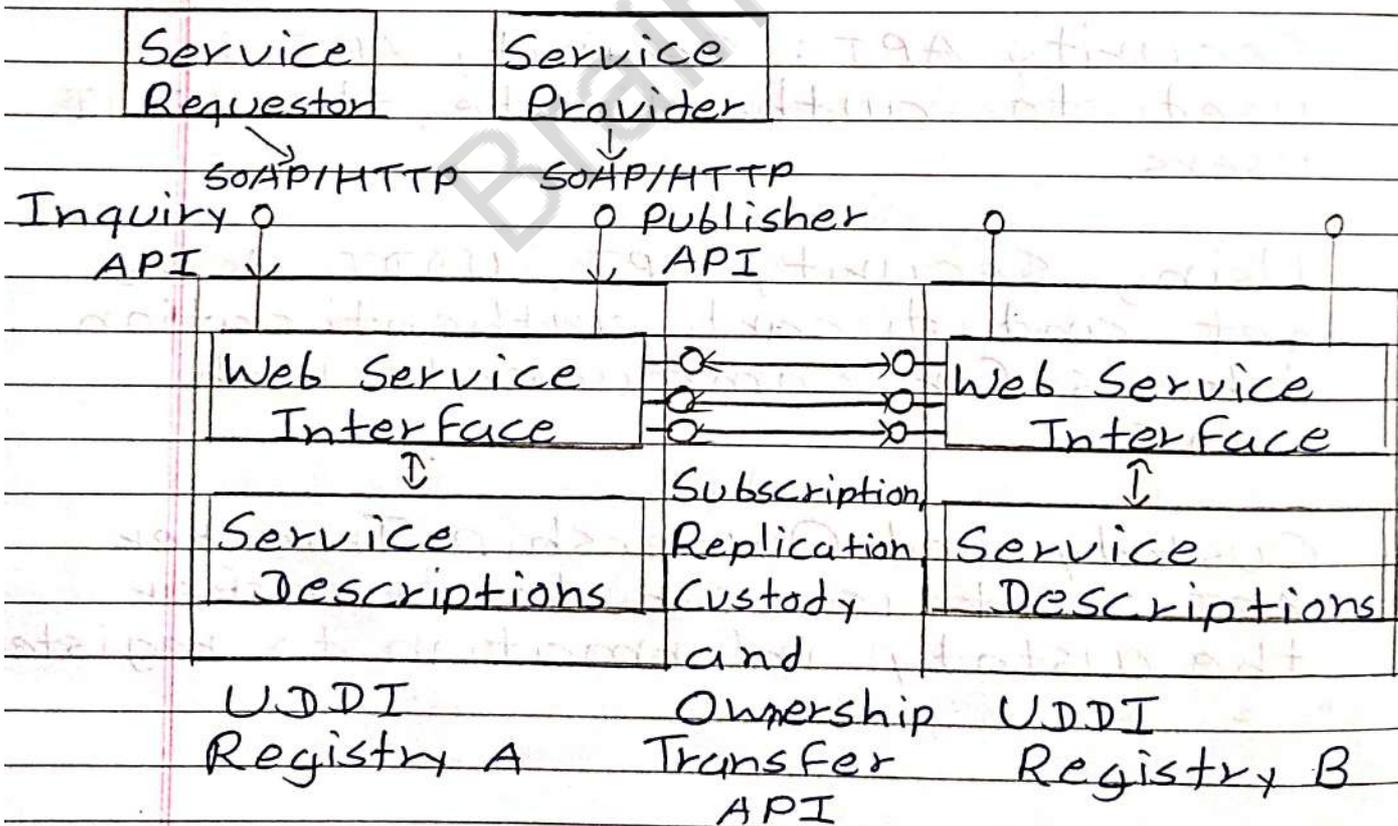
The Green Page contain information about the company's crucial business process.

* Explain UDDI API.

=> There are six types of UDDI API.

- 1) Inquiry API
- 2) Publisher API
- 3) Security API
- 4) Custody and Ownership Transfer API
- 5) Subscription API
- 6) Replication API

UDDI API is used to interact with two UDDI Registries.



1 Inquiry API : Inquiry API is used by service requestor to find the service information.

Inquiry API Provides interface between service requestor and web service interface.

2 Publisher API : Publisher API is used by service provider to provide the service.

Using Publisher API, Service Provider can add, modify and delete the entries in the ~~set~~ registry.

3 Security API : Security API is used to authenticate the UDDI users.

Using Security API, UDDI users get and discard authentication tokens for communicate with UDDI.

4 Custody and Ownership Transfer API : It is used to transfer the custody information to registry.

It is also used to transfer ownership between one publisher to another publisher.

5 Subscription API: It is used to enable the monitoring of changes in registry.

It is used to perform subscribing to track new, modified and deleted entries.

6 Replication API: It is used to support replication of information between registries.

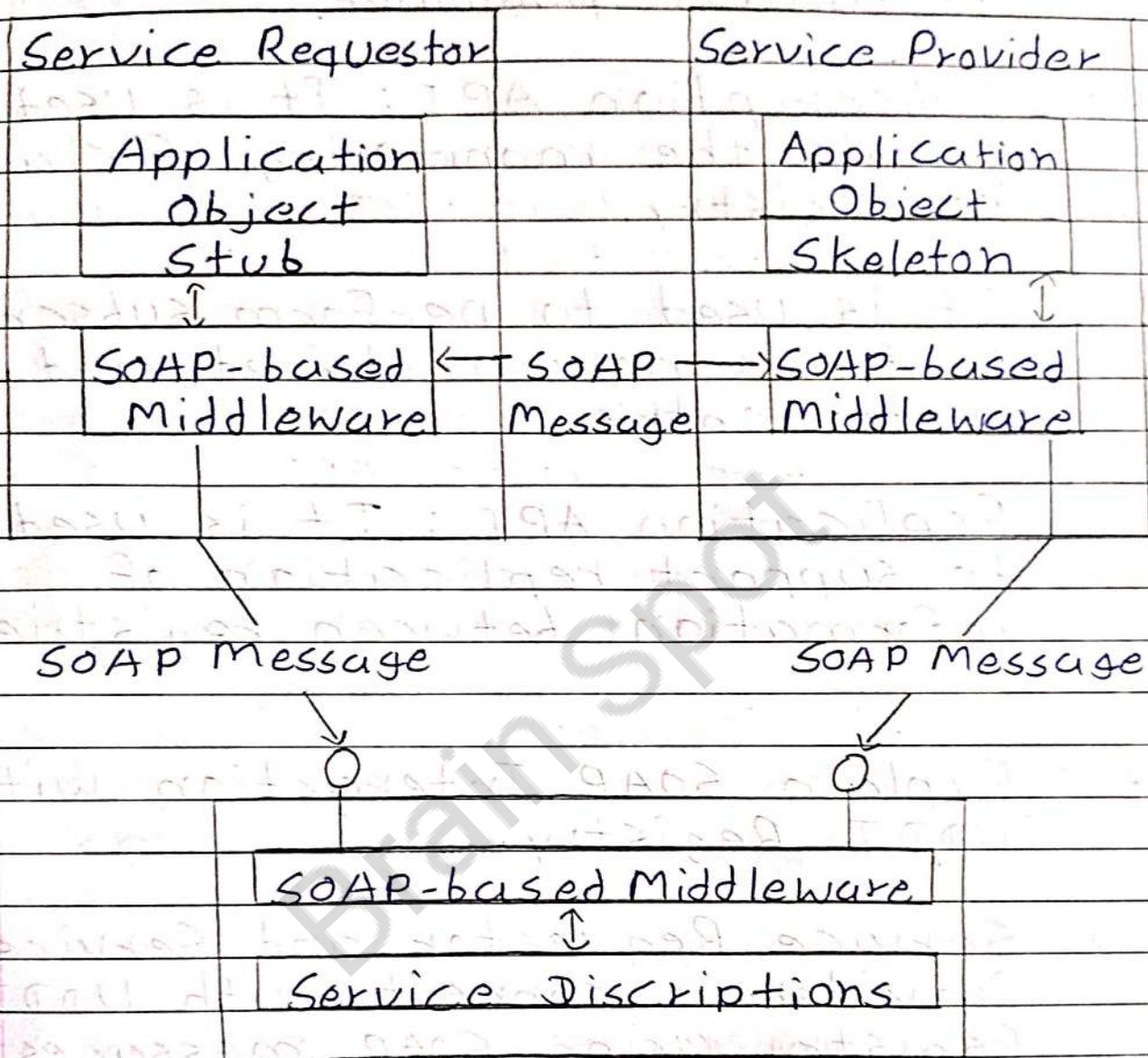
↳ Explain SOAP Interaction with UDDI Registry.

⇒ Service Requestor and Service Provider is connect with UDDI Registry using SOAP messages.

Service Requestor can communicate with UDDI Registry using the SOAP messages interface.

For SOAP messages, there is one SOAP-based middleware for

Provides SOAP Messages



UDDI Registry

Service Requestor can request the UDDI Registry to provides services which is provide by services Provider.

Using SOAP Message, Service Requestor get the information from UDDI Registry.

Service Provider can request the UDDI Registry to add services which is consume by Services Requestor.

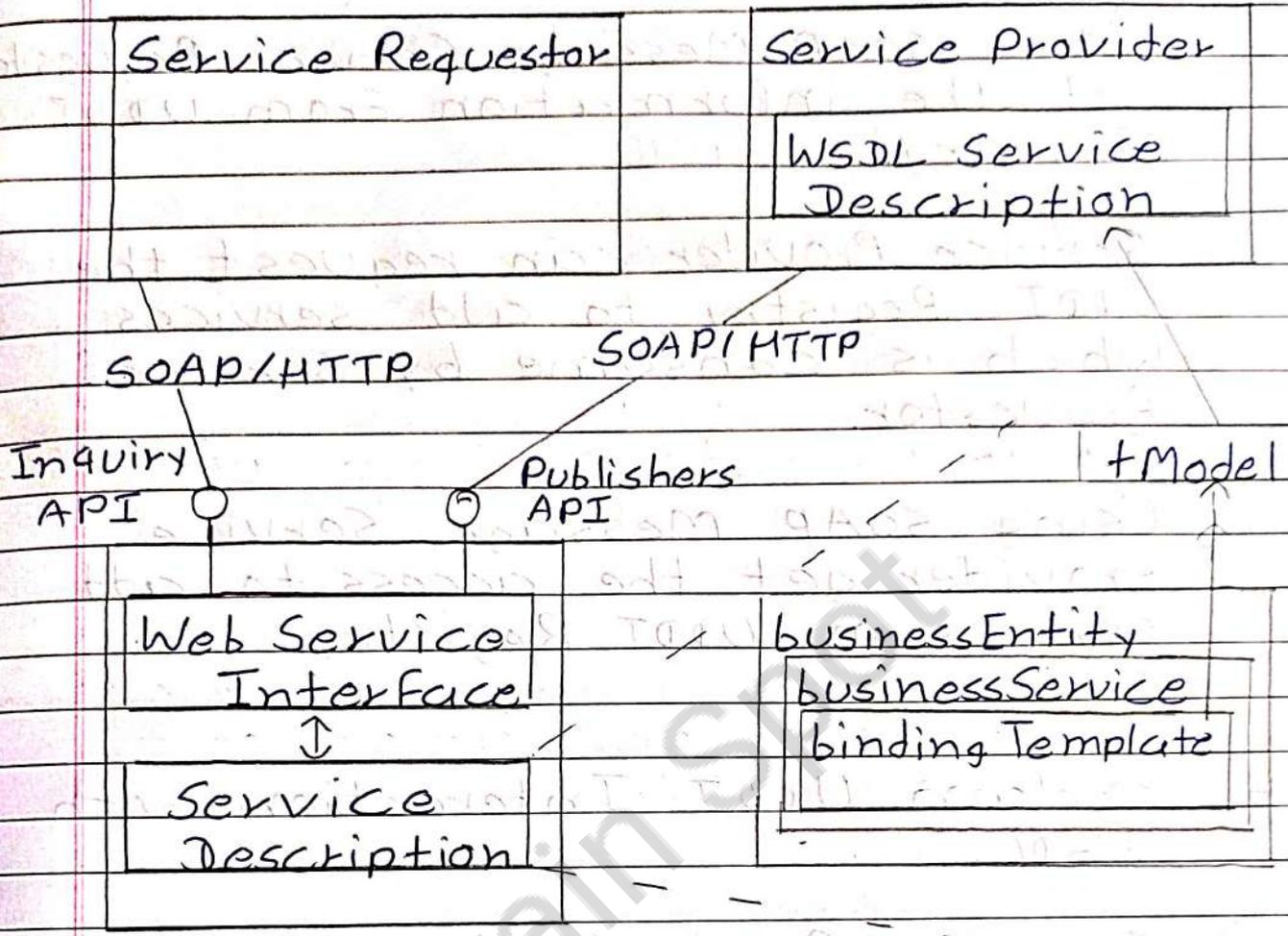
Using SOAP Message, Service Provider get the access to add service in UDDI Registry.

* Explain UDDI Interaction with WSDL.

=> Service Requestor, Using Inquiry API can get information from the UDDI Registry via SOAP or HTTP Interface.

Service Provider can Provides WSDL Service descriptions.

Service Provider, Using Publisher API publish information from the UDDI Registry via SOAP or HTTP Interface.



Web Service Interface Provides information to Service Requestor and Service Provider.

UDDI Registry's tModel is connected with Service Provider's WSDL Service Description.

Using UDDI Registry, Web Service interface, Service Requestor can get the information.

Service Requestor get the service information which is provide by service provider in WSDL Service Description.

Brain Spot