



Technology Enablers

Homework 3

Adv. & Distributed Programming Paradigms

By Asmaa Dalil

Contents

I. Technologies used.....	3
1. Service definition language.....	3
2. Protocol	3
3. Provider programming language.....	3
4. Tool:.....	3
5. Consumer programming language	3
II.Process:	4
1. Development Approach	4
2. Provider:	4
3. Consumer:.....	4

I. Technologies used

1. Service definition language

WSDL (Web Services Description Language).

The WSDL document specifies both the location and the methods and the location of the service of the service. WSD file is an XML format to describe network services as a set of endpoints. In other words, WSDL is designed to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate.

2. Protocol: SOAP (Simple Object Access Protocol) over HTTP

SOAP is a platform and language independent. It is an messaging protocol that is XML-based. And it allows sending and receiving messages, enabling remote procedure calls to be transported via HTTP.

3. Provider programming language:) Java

Java API for XML Web Services (JAX-WS): a standardized API for creating and consuming SOAP web services.

4. Tool: wsgen

For this Homework we use Wsgen: The wsgen tool reads a web service endpoint and generates all the artifacts required to deliver and invoke the web service. This tool is used to generate his WSDL document and skeleton code on the provider side.

5. Consumer programming language

The SOAP module allows us to access web services using SOAP. To create a new SOAP client, we can use the `soap.createClientAsync()` function, which takes the WSDL URL as an argument. In our homework, the URL specifies the location of the file.

Node.js is a JavaScript runtime environment that allows us to run JavaScript code (such as `consumer.js`) outside of a web browser.

II. Process:

1. Development Approach : Code-first

2. Provider:

I used the code-first approach to build this project based on XML/SOAP Web Services, beginning with creating the service business implementation in the target programming language (Java). The creation of the Java server stub and the WSDL file came next.

```
import javax.xml.ws.*;  
  
@WebService  
public class RemoteAccess{  
      
}
```

3. Consumer:

To develop the consumer code, I used JavaScript and the SOAP module to connect to the web service we created (our provider). The SOAP module automatically generates a client stub by parsing the WSDL document that was generated on the provider side. This client stub includes information about the location of the service and the methods available for use. The client stub is then used to execute remote procedure calls.