CSC 3374

جامعة الأخوين

AL AKHAWAYN
UNIVERSITY

Technology Enablers

Homework 2

Adv. & Distributed Programming Paradigms

By Asmaa Dalil

Contents

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 wors, 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 (version: 1.8.0_281)

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: Python (version 3.9.2)

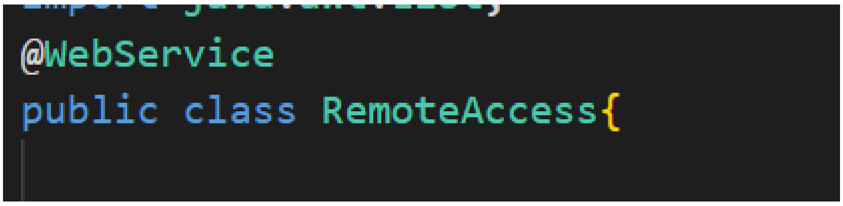
Zeep module: Zeep examines the WSDL file, produces the necessary code to use the service, and then types the file in. While the consumer code is being executed, the client stub code will be produced dynamically.

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.



```
@WebService
public class RemoteAccess{
```

3. Consumer:

By parsing the WSDL document created on the provider side and specifying the location of the service and its methods, I created the consumer code in Python and used the Zeep module in Python to produce the client stub. Using the client stub, remote procedure calls are carried out.