CSC 3374

Design Document

Homework 2

Adv. & Distributed Programming Paradigms

Contents

I. Introduction

After designing the client/server communication in the first assignment. In this regard, we had to build a protocol in parallel that controls the communication between these two parties as well as handle connections between client and server using sockets and manage streams, However, In the second part of the course we were introduced to PRC that overcomes the limitations of the that paradigm. RPC provides us with the luxury of connecting two services and invoking the remote business implementation as if it were local.

II. Project Objectives

In this project, the RAT application utilizes RPC to let the user manipulate a distant system by calling remote functionalities and implementations as though they were local. The client will then be able to get the running processes, take screenshots, or reboot the remote system.

III. User Manual:

1. Service Provider:

The service must be started by the user. The following steps followed:

1. Opening the terminal
2. Changing the directory to the gradle project folder
3. Building the project using the command: gradle build or ./gradlew build
4. Running the project using the command: gradle run or ./gradlew run

2. Service Consumer:

The menu is displayed after the user launches the application offering different choices for the user. This is done through the following steps:

1. Opening the command line

2. Changing the directory to where the consumer code is saved (Consumer.py)

3. Running the application using the command: python RConsumer.py

4. The user is asked to enter a choice from the displayed menu:

   - 1: Get the list of processes running in the remote system
   - 2: Take screenshots of the remote system
   - 3: Reboot the system
   - 4: Exit the program

IV. Application Programming Interface (API)

The API is described using WSDL – Web Service Definition Language which defines the following prototypes of the methods/functions that the service provider offers and the service consumer can invoke as if they were local methods:

Operations:

- GetCapture (): return xsd: base64Binary

  if the operation done successfully then it will return the screenshot, otherwise it will return null

- restart (): return xsd:Boolean

  if the opretion done successfully then it will return true, otherwise it will return false

- getRunningProcess() : return String[]

o In the case it runs successfully, it returns “true” as a Boolean value. Otherwise, it returns “false”.

V. App Description

1. Service Provider:

On this side, we first developed the actual and practical use of the defined functions and methods in the API (applying the code-first approach). The matching tool is then used to generate the WSDL document and the skeleton code.

The service is publicized and prepared to respond to requests from users after the service provider has been successfully launched. The programmer is not made aware of the specifics of how the connection is made or how parameters are transferred. The server stub or skeleton, which is also the one to use the service business implementation by sending the unmarshalled parameters and getting the result to be marshalled, does the parameters unmarshalling and result marshalling.

2. Service Consumer:

The customer, who will use the service provided by the service provider, is on this side. It is only aware of the functions/methods that are prototypes and cannot access the actual business implementation. The client stub/proxy function, a fake function that creates the appearance that a distant business implementation is being executed locally, is used to call the service. Additionally, result unmarshalling and parameter marshalling fall under the purview of the client stub.

VI. Conclusion

Compared to the first assignment, this one allowed us to experience the luxury of invoking functions. This allowed me to practice and see the significance of the integration paradigm towards the end of this assignment. As a result, customers won't be concerned with how the function is implemented on the provider's end of things; instead, costumer calls it as if it were local.

VII. Resources

https://codescracker.com/java/program/java-program-shutdown-computer.html

https://stackoverflow.com/questions/4490454/how-to-take-a-screenshot-in-java

https://community.oracle.com/tech/developers/discussion/3988026/write-a-java-application-that-allows-users-to-restart-windows-services

https://stackoverflow.com/questions/54686/how-to-get-a-list-of-current-open-windows-process-with-java

https://stackoverflow.com/questions/21282327/restart-a-java-application-for-linux-and-windows