

REMOTE PROCEDURE CALL FOR SORTING AN ARRAY - XMLRPC

Aim:

To Implement an XML RPC code for sorting array

Algorithm:

Server side:

1. Import ``SimpleXMLRPCServer`` and ``SimpleXMLRPCRequestHandler`` from the ``xmlrpc.server`` module.
2. Create a custom ``RequestHandler`` class that extends ``SimpleXMLRPCRequestHandler`` and set ``rpc_paths`` to ``('/RPC2',)``.
3. Instantiate the server with ``localhost`` and port ``8000``, using the custom ``RequestHandler``.
4. Register introspection functions to allow clients to discover available methods.
5. Define the ``sort_array`` function that checks if all elements in the array are numbers and sorts the array.
6. Register the ``sort_array`` function with the server, associating it with the name ``'sort_array'``.
7. Start the server with ``serve_forever()``, making it listen for and handle incoming XML-RPC requests.

Client side:

1. Import ``xmlrpc.client`` to interact with the XML-RPC server.
2. Define the server URL (``http://localhost:8000``) and create a ``ServerProxy`` object to communicate with the server.
3. Prompt the user to input a list of numbers separated by spaces.
4. Attempt to convert the input string into a list of floating-point numbers, and handle any ``ValueError`` if the input is invalid.
5. If the input is invalid, print an error message and exit the program.
6. Call the ``sort_array`` function on the server proxy with the list of numbers.
7. Print the sorted array returned from the server.

Program:

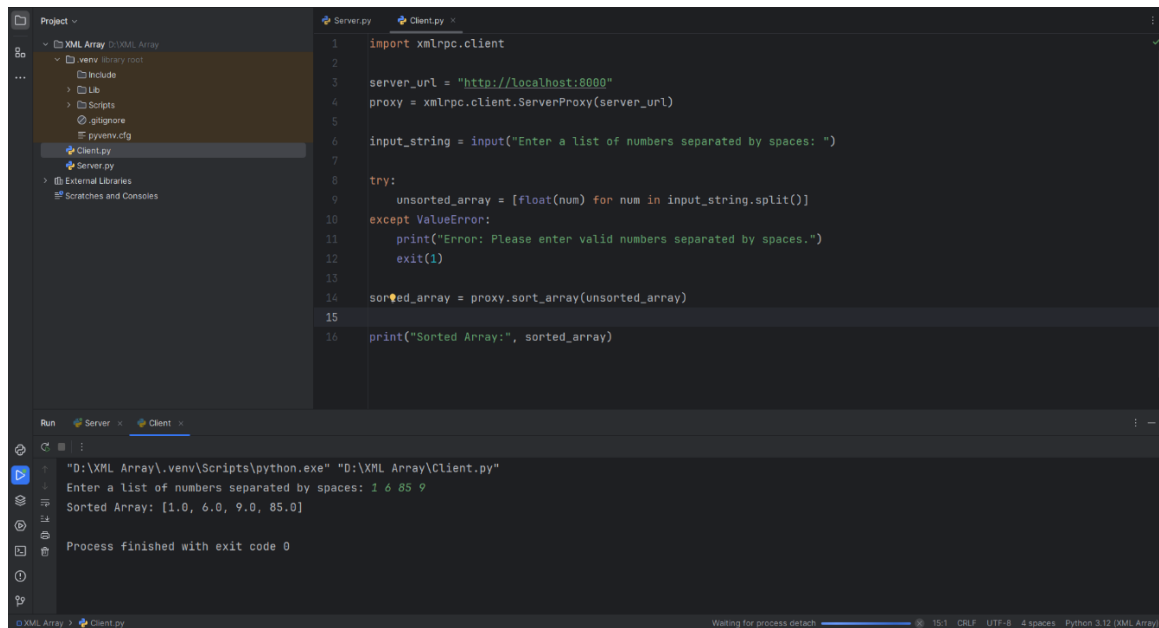
Server side:

```
from xmlrpc.server import SimpleXMLRPCServer
from xmlrpc.server import SimpleXMLRPCRequestHandler
class RequestHandler(SimpleXMLRPCRequestHandler):
    rpc_paths = ('/RPC2',)
server = SimpleXMLRPCServer(('localhost', 8000), requestHandler=RequestHandler)
server.register_introspection_functions()
def sort_array(arr):
    if not all(isinstance(i, (int, float)) for i in arr):
        raise ValueError("All elements of the array must be numbers.")
    return sorted(arr)
server.register_function(sort_array, 'sort_array')
print("Server is running on http://localhost:8000")
server.serve_forever()
```

Client side:

```
import xmlrpc.client
server_url = "http://localhost:8000"
proxy = xmlrpc.client.ServerProxy(server_url)
input_string = input("Enter a list of numbers separated by spaces: ")
try:
    unsorted_array = [float(num) for num in input_string.split()]
except ValueError:
    print("Error: Please enter valid numbers separated by spaces.")
    exit(1)
sorted_array = proxy.sort_array(unsorted_array)
print("Sorted Array:", sorted_array)
```

Output:



The screenshot displays a Python IDE with a project named 'XML Array'. The left sidebar shows the project structure, including a 'venv' directory and a 'Scripts' folder. The main editor window shows the 'Client.py' file with the following code:

```
1 import xmlrpc.client
2
3 server_url = "http://localhost:8000"
4 proxy = xmlrpc.client.ServerProxy(server_url)
5
6 input_string = input("Enter a list of numbers separated by spaces: ")
7
8 try:
9     unsorted_array = [float(num) for num in input_string.split()]
10 except ValueError:
11     print("Error: Please enter valid numbers separated by spaces.")
12     exit(1)
13
14 sorted_array = proxy.sort_array(unsorted_array)
15
16 print("Sorted Array:", sorted_array)
```

The bottom panel shows the 'Run' output for the 'Client' process. The command executed is:

```
"D:\XML Array\venv\Scripts\python.exe" "D:\XML Array\client.py"
```

The output shows the user input and the resulting sorted array:

```
Enter a list of numbers separated by spaces: 1 6 85 9
Sorted Array: [1.0, 6.0, 9.0, 85.0]
```

The process finished with exit code 0.

Result:

Thus, the sorting of array using Remote Procedure Call was executed.