

EX NO:11.D  
231901010

Date:28.10.24  
Hemanth kumar.A

## REMOTE PROCEDURE CALL FOR SORTING AN ARRAY -XMLRPC

### AIM:

To Implement an XML RPC code for sorting array

Algorithm:

### Algorithm

### Server Program

1. **Import Required Library:** Import the xmlrpc.server library, which provides functionalities for setting up an XML-RPC server.
2. **Define sort\_array Function:**
  - Define a function called sort\_array that takes a list (arr) as input.
  - Inside this function, return the sorted version of arr.
3. **Set Up Server:**
  - Create an instance of SimpleXMLRPCServer, specifying localhost as the host and 8000 as the port.
  - Print a message indicating the server is running.
4. **Register Function:**
  - Register the sort\_array function with the XML-RPC server under the name "sort\_array". This makes it available for remote calls.
5. **Run the Server:**
  - Call server.serve\_forever() to start the server and keep it running indefinitely, waiting for client requests.

### Client Program

1. **Import Required Library:** Import the xmlrpc.client library to enable communication with the XML-RPC server.
2. **Connect to Server:**
  - Create a ServerProxy object to connect to the server running on http://localhost:8000.
3. **Prepare Input Array:**

- Define an array `arr` with some integer values that will be sent to the server for sorting.
- 4. **Call `sort_array` Method:**
  - Use the `server.sort_array()` function to send `arr` to the server for sorting.
  - Store the result (sorted array) in a variable `sorted_arr`.
- 5. **Display Results:**
  - Print the original `arr` and the `sorted_arr` to verify the sorting operation was successful.

### **Program:**

#### **SERVER PROGRAM:**

```
import xmlrpc.server
```

```
# Function to sort the
```

```
array def sort_array(arr):
```

```
    return sorted(arr)
```

```
# Setting up the server with
```

```
xmlrpc.server.SimpleXMLRPCServer(('localhost', 8000)) as server:
```

```
    print("Server is running...")
```

```
# Registering the sort_array function as an XML-RPC method
```

```
    server.register_function(sort_array, 'sort_array')
```

```
    server.serve_forever()
```

#### **CLIENT PROGRAM:**

```
import xmlrpc.client
```

```
# Connect to the server server =
```

```
xmlrpc.client.ServerProxy('http://localhost:8000')
```

```
# Input array to be sorted
```

```
arr = [12, 4, 5, 3, 7, 2]
```

```
# Calling the sort_array method on the server  
sorted_arr = server.sort_array(arr)
```

```
# Displaying the result  
print("Original Array:", arr)  
print("Sorted Array:", sorted_arr)
```

## Output

### SERVER OUTPUT

```
"C:\Users\Kaviya V\PycharmProjects\pythonProject2\.venv\Scripts\python.exe"  
Server is running...
```

### CLIENT OUTPUT

```
"C:\Users\Kaviya V\PycharmProjects\pythonProject2\.venv\Scripts\python.exe"  
Original Array: [12, 4, 5, 3, 7, 2]  
Sorted Array: [2, 3, 4, 5, 7, 12]  
  
Process finished with exit code 0
```

## RESULT:

Procedure call for sorting an array - XMLRPC remoted.