XML-RPC: Távoli Eljáráshívás Egyszerűen

Az ***XML-RPC*** egy egyszerű régi protokoll, amely lehetővé teszi eltérő rendszerek közötti adatkommunikációt és a távoli eljáráshívást. Az XML-RPC használatával a fejlesztők úgy tudnak távoli eljárásokat használni egy távoli szerveren, mintha azok helyiek lennének.

Működése

1. **Adatkódolás XML-ben:** Az XML-RPC az adatokat XML formátumban kódolja.
2. **Kommunikáció HTTP-n keresztül:** Az adatcsere a kliens és a szerver között HTTP-n keresztül történik, ami biztosítja az adatátvitel egyszerűségét.
3. **Strukturált Kérések és Válaszok:** Az XML-RPC összetett kéréseket használ, amelyek tartalmazzák a hívni kívánt eljárás nevét és az átadandó paramétereket, valamint a válaszokat, amelyek tartalmazzák a hívás eredményét vagy egy esetleges hibaüzenetet.

Előnyök

- **Egyszerűség:** Az XML-RPC egyszerűsége miatt könnyen implementálható és használható különböző nyelveken és platformokon.
- **Széleskörű Támogatottság:** Több programozási nyelv és keretrendszer is támogatja az XML-RPC-t, ami elősegíti a különböző rendszerek közötti integrációt.
- **Platformfüggetlenség:** Az XML és HTTP alkalmazásával az XML-RPC platformfüggetlen, lehetővé téve különböző operációs rendszerek és hardverek közötti kommunikációt.

Hátrányok

- **Biztonság:** Mivel az XML-RPC adatokat nyílt szöveges formában küldi, biztonsági rétegek (pl. HTTPS) alkalmazása ajánlott.
- **Teljesítmény:** Az XML formátum miatt az XML-RPC lehet lassabb.

Összefoglalás

Az XML-RPC egy hasznos eszköz különböző rendszerek közötti egyszerű adatcsere és távoli eljárások hívásához. Bár vannak biztonsági és teljesítménybeli problémák, de az egyszerűsége és széles körű támogatottsága miatt továbbra is népszerű.

Mintapélda

Töltsük le a XML-RPC csomagot: https://archive.apache.org/dist/ws/xmlrpc/v1.1/ ról az *xmlrpc-1.1.zip*-et

1. Feladat: Tekintsük át a mintakódot és Integer osztály helyett adjunk át összetett adattípust. Pl. class Person(name, age)….

2. Feladat: Írjuk meg a klienst Python-ban vagy valamilyen más nyelven

Kliens:

```
    import java.util.Vector;
    import org.apache.xmlrpc.XmlRpcClient;
    public class Client {
       public static void main (String [] args) {
          try {
             XmlRpcClient client = new
XmlRpcClient("http://localhost/RPC2");
             Vector params = new Vector();
             params.addElement(new Integer(17));
             params.addElement(new Integer(13));
             Object result = client.execute("sample.sum", params);
             int sum = ((Integer) result).intValue();
             System.out.println("The sum is: "+ sum);
          } catch (Exception exception) {
             System.err.println("JavaClient: " + exception);
          }
       }
    }
```

Server:

```
    import org.apache.xmlrpc.WebServer;
    public class Server {
        // this function will be called
        public Integer sum(int x, int y) {
            return new Integer(x + y);
        }
        public static void main(String[] args) {
            try {
                WebServer server = new WebServer(80);
                server.addHandler("sample", new Server());
                server.start();
                System.out.println("Server started.");
            } catch (Exception exception) {
                System.err.println("JavaServer: " + exception);
            }
        }
    }
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:informacios_rendszerek_integralasa:xml-rpc_minta_feladat

Last update: **2024/03/21 20:52**