

## Beispiel 1: Einfache RPC-Anwendung

Im ersten Beispiel wird vom Client auf die übliche Art und Weise eine Socket-Verbindung zum RPC-Server aufgebaut. Nachdem die Input-/Output-Streams des Sockets für die Objektübertragung vorbereitet sind, werden die für den korrekten Methodenaufruf auf dem Server benötigten Informationen versendet. Dabei wird der **Name der Methode** als String verschickt. Es gibt keine Übergabeparameter. Der **Rückgabewert** des RPC-Methodenaufrufs wird als String empfangen und ausgegeben.

Der Quellcode (ZIP) mit Batch-Datei zum Kompilieren und Starten ist auf der Webseite erhältlich ([RPC\\_Example1.zip](#)) ◀ [hier klicken](#)

### Client

```
// ClientRPC01a.java
// Client für Remote Procedure Call

import java.io.*;
import java.net.Socket;

////////////////////////////////////

public class ClientRPC01a {

    public static void main (String args[]) {

        try {

            Socket server = new Socket ( "", 9999 );

            System.out.println ("RPC-Client kontaktiert Server " +
                               server.getInetAddress());

            ObjectOutputStream os = new ObjectOutputStream (
                               server.getOutputStream());

            ObjectInputStream is = new ObjectInputStream (
                               server.getInputStream());

            // Name der aufzurufenden Servermethode senden:
            os.writeObject("getName");
            os.flush();

            // Returnwert der aufgerufenen Servermethode empfangen
            String returnWert = (String)is.readObject();

            // Ergebnis anzeigen:
            System.out.println("Server antwortet: "+returnWert);

            System.in.read();

        } catch (Exception e) {
            System.out.println ( "Keine Verbindung...\nFehler: " + e );
        }
    }
}
```

## RPC-Server

Beim RPC-Server wird ein Server-Socket bereit gestellt, der an dem angegebenen Port auf einen anfragenden Client wartet (**accept**). Der dann erzeugte Arbeits-Socket ist für die Kommunikation und die Übertragung der Object-Streams zwischen Client und Server zuständig. Nachdem der Methodenname als String empfangen wurde, wird er verglichen mit der vorhandenen Methode und ausgeführt. Der Rückgabewert wird als String an den Client zurückgeschickt.

Bei diesem einfachen Beispiel wird eine **statische Methode** aufgerufen. Deshalb muss auch kein Objekt erzeugt werden, um die Methode aufzurufen.

```
// ServerRPC01a.java
// Server für Remote Procedure Call

import java.io.*;
import java.net.Socket;
import java.net.ServerSocket;

class ServerRPC01a {

    public static void main (String args[]) throws IOException {

        ServerSocket ss = new ServerSocket ( 9999 );
        System.out.println("Server erfolgreich gestartet!\n"+
                           "Warte auf Client... \n");
        Socket socket = ss.accept();
        System.out.println( "Client angemeldet...");

        try {

            ObjectOutputStream os =
                new ObjectOutputStream (socket.getOutputStream());

            ObjectInputStream is =
                new ObjectInputStream (socket.getInputStream());

            String methodenName = (String)is.readObject();
            String returnWert  = "Methode nicht gefunden";

            if(methodenName.equals("getName"))
                returnWert = getName( );

            os.writeObject(returnWert);
            is.close();
            os.close();
        } catch(Exception e) {
            System.out.println(e);
        }
    }
    ////////////////////////////////////////
    // angebotene RPC-Methode

    public static String getName() {

        return
            "Dies ist der Rueckgabewert der aufgerufenen RPC-Methode";
    }
}
```