

LogisticaTRUCKServer- I 距離計算サーバ ソケット通信による Javaでの利用例

LogisticaTRUCKServer- I 距離計算サーバAPI ソケット通信によるJava のサンプルプログラム

サンプルプログラム

ポート番号は 44965、「条件値(常時"1")、起点、終点」を送信して「計算条件、起点、終点、総距離 M、True/False、起点認識住所、終点認識住所、起点位置、終点位置、直線距離 M」を受信します。正常に距離計算ができた場合は True、条件値不正・起点終点住所不明などで距離計算できない場合は False を返します。

サンプルプログラムの内容

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;
import java.net.SocketException;

public class SocketSample01 {
    // public static void main(String[] args) throws IOException {
    public static void main(String[] args) {
        try {
            String strServer = "192.139.11.4"; // ← サーバの IP アドレスを指定する
            int intServerPort = 44965;
            // ソケット作成
            Socket socket = new Socket(strServer, intServerPort);
            System.out.println("Connected to Server");
            InputStream in = socket.getInputStream();
            OutputStream out = socket.getOutputStream();
            // while (1 == 1) {
            // 送信する
            String strSoushin = "1,11214,13201";
            // String strSoushin = "1,埼玉県春日部市,大阪市北区";
            byte[] byteSoushinBuffer = strSoushin.getBytes();
            out.write(byteSoushinBuffer);
            // 距離計算結果を受信する
            int intLength = 4096;
            int intReceived;
            byte[] byteJushinBuffer = new byte[4096];
            if ((intReceived = in.read(byteJushinBuffer, 0, intLength)) == -1)
                throw new SocketException("Connection closed prematurely");
            System.out.println("Received: "
                + new String(byteJushinBuffer, 0, intReceived));

            // Split
            String strS = new String(byteJushinBuffer, 0, intReceived);
            String[] strStrings = strS.split(",", -1);
            System.out.println("[Length]" + strStrings.length);
            for (int i = 0; i < strStrings.length; i++) {
                System.out.println("[ " + i + "]" + strStrings[i]);
            }
            socket.close();
        } catch (SocketException e) {
            System.err.println("Socket Error");
            System.exit(-1);
        } catch (IOException e) {

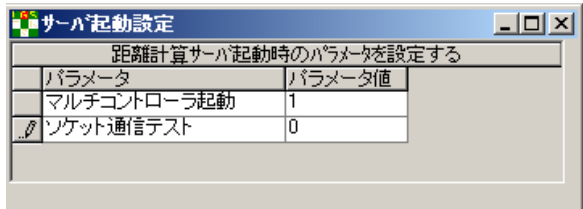
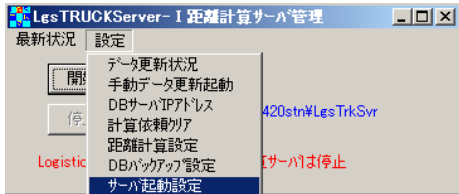
```

```

        System.err.println("IO Error");
        System.exit(-1);
    }
}
}
}

```

サーバ側の起動設定



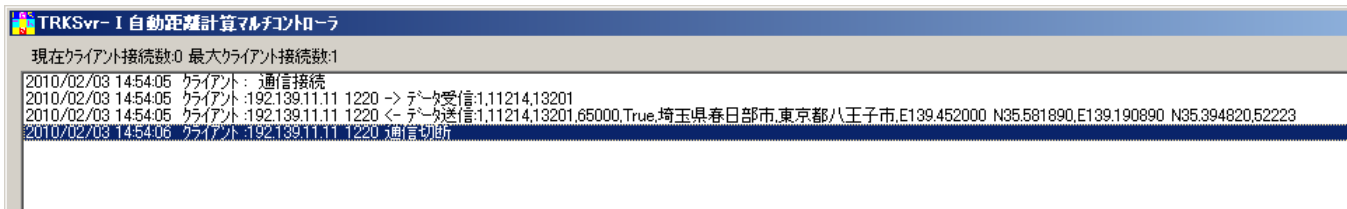
実行結果

```

E:¥Temp¥Java>java SocketSample01
Connected to Server
Received: 1,11214,13201,65000,True,埼玉県春日部市,東京都八王子市,E139.452000 N35.581890,E139.190890 N35.394820,52223
[Length]10
[0]1
[1]11214
[2]13201
[3]65000
[4]True
[5]埼玉県春日部市
[6]東京都八王子市
[7]E139.452000 N35.581890
[8]E139.190890 N35.394820
[9]52223

```

距離計算サーバのモニタ画面



参考

Java クラス Socket については
<http://sdc.sun.co.jp/java/docs/j2se/1.4/ja/docs/ja/api/java/net/Socket.html>
 を参照願います。