

LogisticaTRUCKServer- II 距離計算サーバ API ソケット通信 Javaでのソケット通信**Javaでのソケット通信の実行****サンプルプログラム**

ポート番号は 44963、「条件値,起点,終点」を送信して、「条件値,起点,終点,総距離(m),有料道距離(m),総時間(秒),有料道時間(秒),TrueまたはFalse,起点認識住所,終点認識住所,起点位置,終点位置」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。

```
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) {
        try {
            String strServer = "192.139.11.6";
            int intServerPort = 44963;
            // 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();
            String strSoushin = "272,東京都千代田区大手町 1-1-1,大阪市北区中之島 1-1-2";
            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);
        }
    }
}
```

実行結果

```
コマンドプロンプト
E:\Program Files\lgsPAk01AX01\SocketSample\Java>java SocketSample01
Connected to Server
Received: 272,東京都千代田区大手町1-1-1,大阪市北区中之島1-1-2,512444,0,91781,0,Tr
ue,東京都千代田区大手町1丁目1番地1号,大阪府大阪市北区中之島1丁目1番地2号,
E139.455684 N35.410066,E135.303372 N34.412303
[Length]12
[0]272
[1]東京都千代田区大手町1-1-1
[2]大阪市北区中之島1-1-2
[3]512444
[4]0
[5]91781
[6]0
[7]True
[8]東京都千代田区大手町1丁目1番地1号
[9]大阪府大阪市北区中之島1丁目1番地2号
[10]E139.455684 N35.410066
[11]E135.303372 N34.412303
```

動作を確認した環境

WindowsXP/SP2

Java Version6 Update11

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。

LogisticaTRUCKServer- II 距離計算サーバ API ソケット通信 Perlでのソケット通信

Perl サンプルプログラム

サンプルプログラム

サンプルプログラムは、C:\Program Files\lgsPAk01AX01\SocketSample\PERL\sample.plにあります。

ポート番号は 44963、「条件値,起点,終点」を送信して、「条件値,起点,終点,総距離(m),有料道距離(m),総時間(秒),有料道時間(秒),TrueまたはFalse,起点認識住所,終点認識住所,起点位置,終点位置」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。

```

E:\Program Files\lgsPAk01AX01\SocketSample\PERL\sample.pl - 秀丸
ファイル(F) 編集(E) 検索(S) ウィンドウ(W) マクロ(M) その他(O) 1:1
#! E:/Perl/bin/perl↓
use Socket;↓
$buf="272,11101,14101";↓
printf("送信[%s]\n",$buf);↓
$proto=getprotobyname("tcp");↓
printf("tcp プロトコル番号は[%s]\n",$proto);↓
# ソケットオープン↓
socket(C_SOCKET,PF_INET,SOCK_STREAM,$proto) || die "接続に失敗しました $!";↓
$ipaddr_bin=inet_aton("192.139.11.6");↓
connect(C_SOCKET,pack_sockaddr_in(44963,$ipaddr_bin)) || die "接続に失敗しました $!";↓
$ipaddr_str=inet_ntoa($ipaddr_bin);↓
printf("IPアドレスは[%s]\n",$ipaddr_str);↓
#ソケットのバッファリングはしない↓
select(C_SOCKET);$|=1;select(STDOUT);↓
# 14101(横浜市鶴見区) 11107(さいたま市浦和区)↓
$send="272,14101,11101";↓
printf("送信[%s]\n",$send);↓
print C_SOCKET $send;↓
#↓
sysread(C_SOCKET,$recv,200);↓
printf("受信[%s]\n",$recv);↓
#sleep(1);↓
↓
# 〒100-0004(東京都千代田区丸の内) ↓
$send="272,1000004,大阪市北区中之島1-1-28";↓
printf("送信[%s]\n",$send);↓
print C_SOCKET $send;↓
#↓
sysread(C_SOCKET,$recv,200);↓
printf("受信[%s]\n",$recv);↓
#sleep(1);↓
close(C_SOCKET);↓
exit;↓
[EOF]
外部ヘルプ? ファイルを..

```

```

コマンド プロンプト
E:\Program Files\lgsPAk01AX01\SocketSample\PERL>perl sample.pl
送信[272,11101,14101]
tcp プロトコル番号は[6]
IPアドレスは[192.139.11.6]
送信[272,14101,11101]
受信[272,14101,11101,81150,6360,10196,402,True,神奈川県横浜市鶴見区,埼玉県さいたま市西区,E139.410830 N35.301850,E139.330539 N35.551343]
送信[272,1000004,大阪市北区中之島1-1-28]
受信[272,1000004,大阪市北区中之島1-1-28,514148,1200,91917,91,True,東京都千代田区大手町,大阪府大阪市北区中之島1丁目1番地2-8号,E139.460623 N35.410491,E135.303999 N34.412117]
E:\Program Files\lgsPAk01AX01\SocketSample\PERL>

```

動作を確認した環境

WindowsXP/SP2

ActivePerl5.8.7 Build815 (Windows用)

参考

PerlのSocket関数については

<http://www.g-ishihara.com/perl/network01.html>

を参照願います。

LogisticaTRUCKServer- II 距離計算サーバ API ソケット通信 VisualBasicでのソケット通信

VBNET サンプルプログラム

ポート番号は 44963、「条件値,起点,終点」を送信して、「条件値,起点,終点,総距離(m),有料道距離(m),総時間(秒),有料道時間(秒),TrueまたはFalse」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。

Public Class Form1

Inherits System.Windows.Forms.Form

Private intPortNo As Integer = 44963 '← Port 番号

Private objNetStream As System.Net.Sockets.NetworkStream

Private objTcpClient As System.Net.Sockets.TcpClient

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim bytReceiveBuffer(255) As Byte

Dim bytSendBuffer As Byte()

Dim intDataLength As Integer

Dim strReceivedData As String

objTcpClient = New System.Net.Sockets.TcpClient

With objTcpClient

Try

'リモートホストと接続します。

'objTcpClient.Connect("localhost", intPortNo)

objTcpClient.Connect("192.139.11.4", intPortNo)

objNetStream = .GetStream

With objNetStream

'データをリモートホストへ送信します。

bytSendBuffer = System.Text.Encoding.Default.GetBytes(TextBox1.Text)

.Write(bytSendBuffer, 0, bytSendBuffer.Length)

'リモートホストからの返信を受信します。

Do

If .DataAvailable Then

intDataLength = .Read(bytReceiveBuffer, 0, bytReceiveBuffer.Length)

strReceivedData &= System.Text.Encoding.Default.GetString(bytReceiveBuffer).Substring(0, intDataLength)

Elseif Not strReceivedData Is Nothing Then

TextBox2.Text = strReceivedData

Exit Do

End If

Loop

End With

Catch Ex As Exception

'エラーメッセージをメッセージボックスで表示します。

MsgBox(Ex.Message)

End Try

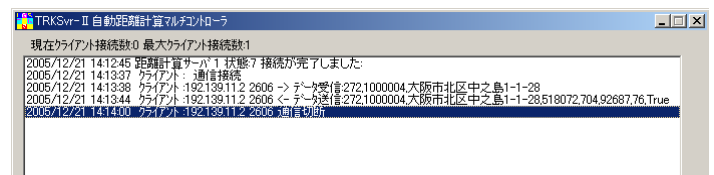
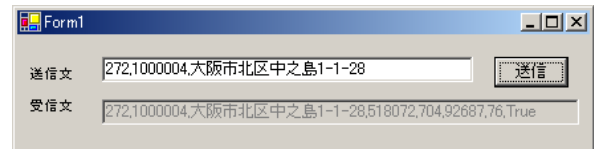
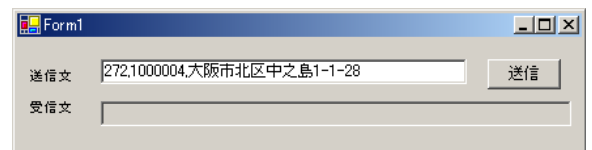
'TCP接続を終了します。

.Close()

End With

End Sub

End Class



LogisticaTRUCKServer- II 距離計算サーバ API ソケット通信 C#でのソケット通信

C# サンプルプログラム

サンプルプログラムは、コンソールアプリケーションで**ポート番号は 44963**、「**条件値,起点,終点**」を送信して、「**条件値,起点,終点,総距離(m),有料道距離(m),総時間(秒),有料道時間(秒),TrueまたはFalse**」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。

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
```

```
namespace MyApp
{
```

```
    /// <summary>
```

```
    /// Class1 の概要の説明です。
```

```
    /// </summary>
```

```
    class Class1
```

```
    {
```

```
        public static void StartClient()
```

```
        {
```

```
            // Data buffer for incoming data.
```

```
            byte[] bytes = new byte[1024];
```

```
            // Connect to a remote device.
```

```
            try
```

```
            {
```

```
                // Establish the remote endpoint for the socket.
```

```
                // The name of the
```

```
                // remote device is "host.contoso.com".
```

```
                IPHostEntry ipHostInfo = Dns.Resolve("192.139.11.4"); //host.contoso.com");
```

```
                IPAddress ipAddress = ipHostInfo.AddressList[0];
```

```
                IPEndPoint remoteEP = new IPEndPoint(ipAddress, 44963);
```

```
                // Create a TCP/IP socket.
```

```
                Socket sender = new Socket(AddressFamily.InterNetwork,
                    SocketType.Stream, ProtocolType.Tcp );
```

```
                // Connect the socket to the remote endpoint. Catch any errors.
```

```
                try
```

```
                {
```

```
                    sender.Connect(remoteEP);
```

```
                    Console.WriteLine("Socket connected to {0}",
                        sender.RemoteEndPoint.ToString());
```

```
                    // Encode the data string into a byte array.
```

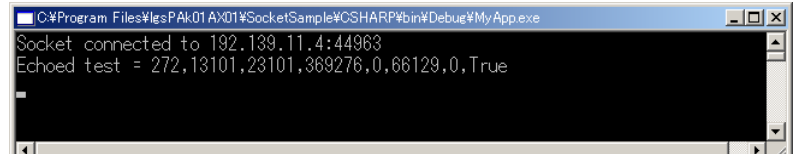
```
                    //byte[] msg = Encoding.ASCII.GetBytes("This is a test<EOF>");
```

```
                    byte[] msg = Encoding.ASCII.GetBytes("272,13101,23101");
```

```
                    // Send the data through the socket.
```

```
                    int bytesSent = sender.Send(msg);
```

実行例



LogisticaTRUCKServer- II 距離計算サーバ API ソケット通信 PHPでのソケット通信

PHP サンプルプログラム

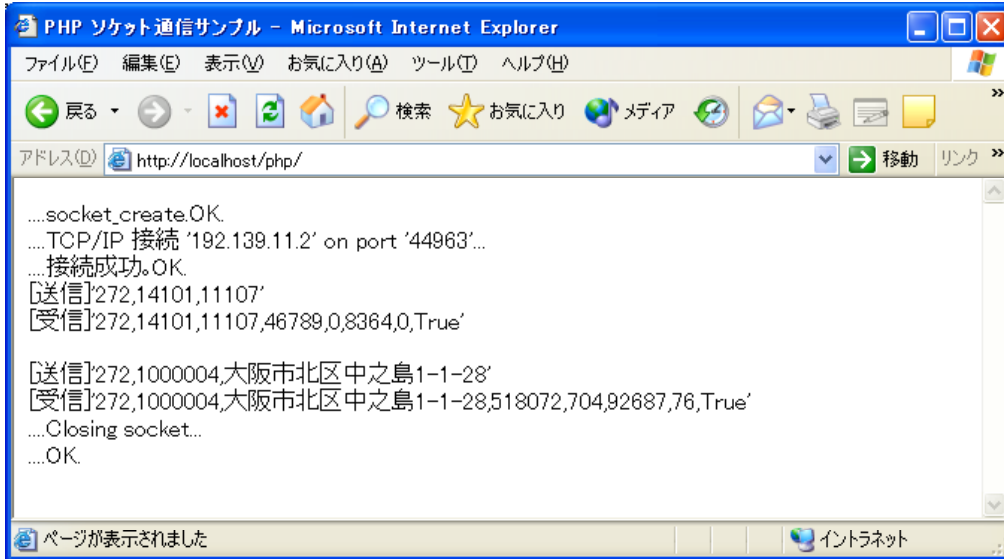
ポート番号は 44963、「条件値,起点,終点」を送信して、「条件値,起点,終点,総距離(m),有料道距離(m),総時間(秒),有料道時間(秒),TrueまたはFalse」を受信します。正常に距離計算ができた場合はTrue,条件値不正・起点終点住所不明などで距離計算できない場合はFalseを返します。

```

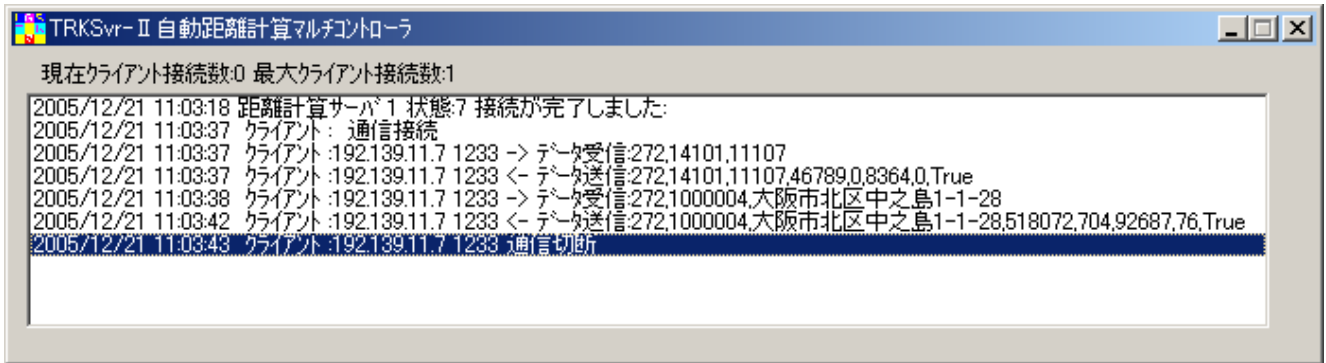
E:\Program Files\Apache Group\Apache2\htdocs\php\index.php [EUC] - 秀丸
ファイル(F) 編集(E) 検索(S) ウィンドウ(W) マクロ(M) その他(O) 3: 32
<HTML>↓
<HEAD>↓
<TITLE>PHP ソケット通信サンプル</TITLE>↓
</HEAD>↓
<BODY>↓
<?php↓
$service_port = 44963; /* ポート番号 */↓
$address = gethostbyname('192.139.11.2'); /* ポート番号アドレス */↓
$socket = socket_create(AF_INET, SOCK_STREAM, 0); /* Create a TCP/IP socket. */↓
if ($socket < 0) {↓
    echo "....socket_create() failed: reason: " . socket_strerror($socket) . "\n";↓
} else {↓
    echo "....socket_create.OK.\n";↓
}↓
print "<BR>";↓
echo "...TCP/IP 接続 '$address' on port '$service_port'...";↓
print "<BR>";↓
$result = socket_connect($socket, $address, $service_port);↓
if ($result < 0) {↓
    echo "....socket_connect()失敗.\nReason: ($result) " . socket_strerror($result) . "\n";↓
} else {↓
    echo "....接続成功.OK.\n";↓
}↓
print "<BR>";↓
$send="272,14101,11107"; // 14101(横浜市鶴見区) 11107(さいたま市浦和区)↓
$recv="";↓
echo "[送信]" . $send . "\n";↓
print "<BR>";↓
socket_write($socket, $send, strlen($send));↓
$recv = socket_read($socket, 2048);↓
echo "[受信]" . $recv . "\n";↓
print "<BR>";↓
sleep(1);↓
print "<BR>";↓
$send="272,1000004,大阪市北区中之島1-1-28"; // 〒100-0004(東京都千代田区丸の内) ↓
echo "[送信]" . $send . "\n";↓
print "<BR>";↓
$send=mb_convert_encoding($send,"SJIS","EUC-JP"); // EUC-JP ->> SJIS↓
$recv="";↓
socket_write($socket, $send, strlen($send));↓
$recv = socket_read($socket, 2048);↓
$recv=mb_convert_encoding($recv,"EUC-JP","SJIS"); // SJIS ->> EUC-JP↓
echo "[受信]" . $recv . "\n";↓
print "<BR>";↓
sleep(1);↓
echo "....Closing socket...";↓
print "<BR>";↓
socket_close($socket);↓
echo "....OK.";↓
print "<BR>";↓
?>↓
</BODY>↓
</HTML>↓
外部へ... ファイルを... 次の秀... ダイレクト...

```

PHP 実行結果



PHP 距離計算サーバのモニタ画面



PHP 動作を確認した環境

WindowsXP/SP2
 Apache HTTP Server 2.0.55 (Windows用)
 PHP 5.1.1 (Windows用)

PHP 参考

Windows 用 PHP 拡張モジュールで、Socket関数を使用するためのPHPの設定については
<http://php.morva.net/manual/ja/install.windows.extensions.php>
 を参照願います。