

QTcpSocket

TCP (Transmission Control Protocol) is a reliable, stream-oriented, connection-oriented transport protocol. It is especially well suited for continuous transmission of data.

QTcpSocket

TCP (Protokół Kontroli Przesyłu) jest niezawodnym, zorientowanym strumieniowo, zorientowanym połączeniowo protokołem transportowym. Jest szczególnie przydatny do ciągłego przesyłu danych.

QTcpSocket - klient

Komunikacja z serwerem

QTcpSocket - klient

plik projektu

QT += network widgets

QTcpSocket - klient

plik nagłówkowy klasy

```
QTcpSocket *tcpSocket;
```

QTcpSocket - klient

plik źródłowy klasy

```
tcpSocket = new QTcpSocket(this);
```

QTcpSocket - klient

plik źródłowy klasy c.d.

```
connect(tcpSocket, SIGNAL(readyRead()),  
        this, SLOT(czytajWiadomosc()));
```

QTcpSocket - klient

plik źródłowy klasy c.d.2

```
QDataStream in(tcpSocket);
in.setVersion(QDataStream::Qt_4_0);

if (blockSize == 0) {
    if (tcpSocket->bytesAvailable() < (int)sizeof(quint16))
        return;

    in >> blockSize;
}

if (tcpSocket->bytesAvailable() < blockSize)
    return;

QString message;
in >> message;
```


QTcpSocket - klient

plik źródłowy klasy c.d.3

```
tcpSocket->abort();  
tcpSocket->connectToHost(adres_ip, port);  
tcpSocket->write("informacja");
```

QTcpSocket - serwer

Centralne przetwarzanie komunikatów

QTcpSocket - serwer

plik projektu

QT += network widgets

QTcpSocket - serwer

plik nagłówkowy klasy

```
class MojSerwer : public QTcpServer  
{  
    Q_OBJECT
```

QTcpSocket - serwer

plik nagłówkowy klasy c.d.

```
protected:  
    void incomingConnection(qintptr socketDescriptor);
```

QTcpSocket - serwer

plik źródłowy klasy (*incomingConnection*)

```
QTcpSocket tcpSocket;  
if (!tcpSocket.setSocketDescriptor(socketDescriptor))  
    return;  
  
tcpSocket.waitForReadyRead();  
QString message(tcpSocket.readAll());
```

QTcpSocket - serwer

plik źródłowy klasy (*incomingConnection*)
c.d.

```
QByteArray block;  
QDataStream out(&block, QIODevice::WriteOnly);  
out.setVersion(QDataStream::Qt_4_0);  
  
out << (quint16)0;  
out << "informacja zwrotna";  
out.device()->seek(0);  
out << (quint16)(block.size() - sizeof(quint16));  
  
tcpSocket.write(block);  
tcpSocket.disconnectFromHost();  
tcpSocket.waitForDisconnected();
```

Dziękuję za uwagę

Miłego dnia
:)