



방문해 주셔서 감사합니다~ 프로그래머 여러분에게 도움이 되는 사이트가 되겠습니다.

로그인

회원가입

회원정보찾기

111

ORACLEJAVA Community

실무 프로그래머를 위한 강좌/팁. Since 1997 오라클 강좌



MENU

- Oracle Tip
- Oracle
- Oracle 11g/12c
- [PL/SQL]
- Oracle Hint
- SQL Tuning
- BackUp & Recovery
- Informatica
- Oracle Pro*C
- Unix

- 앵글러(angular) 강좌
- 파이썬(Python)
- 버텍스(Vert.x)
- MiPlatform/XPlatform/GAUCE
- Gradle/Maven
- Apache/Tomcat
 - 일반
 - 질문
 - 답변
- ASP
 - 일반
 - 질문
 - 답변
- PHP/MySQL
 - 일반
 - 질문
 - 답변
- JavaScript/JSON
- C
- C++
- C자료구조
- Linux
- MSSQL
- WebLogic/JEUS
- XML/UML

- 자바 강좌
 - 스프링 | 자바 | 안드로이드 | 오라클
 - Java
 - Java Swing/AWT
 - Java Network
 - Java Algorithm
 - Java JDBC
 - AJAX/jQuery/HTML5
 - Ant/JUnit
 - EJB
 - Eclipse/MyBatis/Hibernate
 - Servlet&JSP
 - Spring Framework&Boot
 - JPA, Spring Data JPA, Querydsl, JPQL, ORM
 - Struts Framework
 - WebService
 - 안드로이드(Android)
 - 아이폰(iPhone)
 - 자바스터디2기
 - 자바스터디1기
 - 일반
 - 질문
 - 답변
 - 평일주간 미니 프로젝트
 - 일반
 - 질문
 - 답변
 - 스터디 지원
 - DRM For P2P 닷넷으로
 - 치과용 솔루션 만들기!
 - TOAD를 만들자!
 - 오프라인교육센터과제
 - 오프라인모집중강좌
 - Q&A (질문은 이곳에 ...)
 - 일반
 - 질문
 - 답변
 - 구인/구직
 - 구인
 - 구직
 - 잡담
 - 홍보게시판
 - IT용어
 - 방명록
 - 공지사항

닷넷 강
작
UI개발기
술
스마트폰
앱개발
기타 강
좌
스터디
프로젝트
오프라인
교육장
기타 게
시판

Network

C#네트워크 SSL TCP 서버, 클라이언트, SSL 기반의 소켓 통신(SslStream)

스누피2 (106.♡.246.18) 0 1,028

C#네트워크 SSL TCP 서버, 클라이언트, SSL 기반의 소켓 통신(SslStream)

2019.01.20 14:58

시작 -> Visual studio 2017 -> 개발자명령프롬프트에서 아래 명령을 실행하여 서버 인증서를 먼저 만들자.
(관리자 권한으로 실행)

1. makecert -sr LocalMachine -ss root -r -n "CN=localhost2" -sky exchange -sk 123456

2. SSL Server

```
using System.Net;
using System.Net.Sockets;
using System.Net.Security;
using System.Security.Authentication;
using System.Security.Cryptography.X509Certificates;
using System.Diagnostics;
using System;
using System.Text;
```

```

public sealed class SslServer
{
    static X509Certificate serverCertificate = null;

    public static void RunServer()
    {
        try
        {
            // makecert -sr LocalMachine -ss root -r -n "CN=localhost2" -sky exchange -sk 123456
            X509Store store = new X509Store(StoreName.Root, StoreLocation.LocalMachine);
            store.Open(OpenFlags.ReadOnly);
            var certificates = store.Certificates.Find(X509FindType.FindBySubjectDistinguishedName, "CN=localhost2", false);

            store.Close();

            if (certificates.Count == 0)
            {
                Console.WriteLine("Server certificate not found...");
                return;
            }
            else
            {
                serverCertificate = certificates[0];
            }

            TcpListener listener = new TcpListener(IPAddress.Any, 8080);
            listener.Start();

            while (true)
            {
                Console.WriteLine("Waiting for a client to connect...");
                Console.WriteLine();

                TcpClient client = listener.AcceptTcpClient();
                ProcessClient(client);
            }
        }
        catch (Exception ex)
        {
            Trace.WriteLine(string.Format("Error : {0}", ex.Message));
        }
    }

    static void ProcessClient(TcpClient client)
    {
        SslStream sslStream = new SslStream(client.GetStream(), false);

        try
        {
            sslStream.AuthenticateAsServer(serverCertificate, false, SslProtocols.Tls, true);

            // Set timeouts for the read and write to 5 seconds.
            //sslStream.ReadTimeout = 5000;
            //sslStream.WriteTimeout = 5000;

            // Read a message from the client.
            Console.WriteLine("Waiting for client message...");
            string messageData = ReadMessage(sslStream);
            Console.WriteLine("Received : {0}", messageData.Substring(0, messageData.IndexOf("$")));

            // Write a message to the client
            messageData = "[reply] " + messageData;
            byte[] message = Encoding.UTF8.GetBytes(messageData);
            sslStream.Write(message);
            Console.WriteLine("Sending hello message");
            Console.WriteLine();
        }
        catch (AuthenticationException e)
        {
            Console.WriteLine("Exception: {0}", e.Message);
            if (e.InnerException != null)
            {
                Console.WriteLine("Inner exception: {0}", e.InnerException.Message);
            }
            Console.WriteLine("Authentication failed - closing the connection.");
            sslStream.Close();
        }
    }
}

```

```

        client.Close();
        return;
    }
    finally
    {
        // The client stream will be closed with the sslStream
        // because we specified this behavior when creating
        // the sslStream.
        sslStream.Close();
        client.Close();
    }
}

static string ReadMessage(SslStream sslStream)
{
    // Read the message sent by the client.
    // The client signals the end of the message using the
    // "$" marker.
    byte[] buffer = new byte[2048];
    StringBuilder messageData = new StringBuilder();
    int bytes = -1;
    do
    {
        // Read the client's test message.
        bytes = sslStream.Read(buffer, 0, buffer.Length);
        // Use Decoder class to convert from bytes to UTF8
        // in case a character spans two buffers.
        Decoder decoder = Encoding.UTF8.GetDecoder();
        char[] chars = new char[decoder.GetCharCount(buffer, 0, bytes)];
        decoder.GetChars(buffer, 0, bytes, chars, 0);
        messageData.Append(chars);

        // Check for EOF or an empty message.
        if (messageData.ToString().IndexOf("$") != -1)
        {
            break;
        }
    } while (bytes != 0);

    return messageData.ToString();
}

}

class Program
{
    static void Main(string[] args)
    {
        SslServer.RunServer();

        Console.WriteLine("Press the any key to continue...");
        Console.ReadLine();
    }
}

```

3. SSL Client

```

using System;
using System.Collections;
using System.Net;
using System.Net.Security;
using System.Net.Sockets;
using System.Security.Authentication;
using System.Text;
using System.Security.Cryptography.X509Certificates;

namespace Examples
{
    public class SslClient
    {
        private static Hashtable certificateErrors = new Hashtable();

        // The following method is invoked by the RemoteCertificateValidationDelegate.
        public static bool ValidateServerCertificate(
            object sender,
            X509Certificate certificate,
            X509Chain chain,

```

```

    SslPolicyErrors sslPolicyErrors)
{
    if (sslPolicyErrors == SslPolicyErrors.None)
        return true;

    Console.WriteLine("Certificate error: {0}", sslPolicyErrors);

    // Do not allow this client to communicate with unauthenticated servers.
    return false;
}

public static void RunClient(string machineName, string serverName)
{
    // Create a TCP/IP client socket.
    // machineName is the host running the server application.
    TcpClient client = new TcpClient(machineName, 8080);
    Console.WriteLine("Client connected.");
    // Create an SSL stream that will close the client's stream.
    SslStream sslStream = new SslStream(
        client.GetStream(),
        false,
        new RemoteCertificateValidationCallback(ValidateServerCertificate),
        null
    );
    // The server name must match the name on the server certificate.
    try
    {
        sslStream.AuthenticateAsClient(serverName);
    }
    catch (AuthenticationException e)
    {
        Console.WriteLine("Exception: {0}", e.Message);
        if (e.InnerException != null)
        {
            Console.WriteLine("Inner exception: {0}", e.InnerException.Message);
        }
        Console.WriteLine("Authentication failed - closing the connection.");
        client.Close();
        return;
    }

    // Encode a test message into a byte array.
    // Signal the end of the message using the "<EOF>".
    byte[] message = Encoding.UTF8.GetBytes("Hello from the client.$");
    // Send hello message to the server.
    sslStream.Write(message);
    sslStream.Flush();
    // Read message from the server.
    string serverMessage = ReadMessage(sslStream);

    Console.WriteLine("Server says: {0}", serverMessage);
    // Close the client connection.
    client.Close();
    Console.WriteLine("Client closed.");
}

static string ReadMessage(SslStream sslStream)
{
    // Read the message sent by the server.
    // The end of the message is signaled using the
    // "<EOF>" marker.
    byte[] buffer = new byte[2048];
    StringBuilder messageData = new StringBuilder();
    int bytes = -1;
    do
    {
        bytes = sslStream.Read(buffer, 0, buffer.Length);

        // Use Decoder class to convert from bytes to UTF8
        // in case a character spans two buffers.
        Decoder decoder = Encoding.UTF8.GetDecoder();
        char[] chars = new char[decoder.GetCharCount(buffer, 0, bytes)];
        decoder.GetChars(buffer, 0, bytes, chars, 0);
        messageData.Append(chars);
        // Check for EOF.
        if (messageData.ToString().IndexOf("<EOF>") != -1)
        {
            break;
        }
    }
}

```

```

    } while (bytes != 0);

    return messageData.ToString();
}
public static void Main(string[] args)
{
    string serverCertificateName = null;
    string machineName = null;

    machineName = "127.0.0.1";
    serverCertificateName = "localhost2"; // tried Test, LocalMachine and 127.0.0.1
    SslClient.RunClient(machineName, serverCertificateName);
    Console.ReadKey();
}
}
}

```

4. 실행결과

[서버]

Waiting for a client to connect...

Waiting for client message...

Received : Hello from the client.

Sending hello message


Waiting for a client to connect...

[클라이언트]

Client connected.

Server says: [reply] Hello from the client.\$

Client closed.



좋아요 0개 공유하기


Comments

로그인한 회원만 댓글 등록이 가능합니다.

[목록](#)

Category

- C#
- C#, ASP.NET
- ADO.NET
- C# Network
- 자마린(Xamarin)
- WPF 기초강좌
- VB.NET

 [프린트](#)

공지사항

+


배너 광고, 홍보 문의 안내	06.26
프리랜서분들 무료 구직 알선! 미리미리 일정 챙기세요~	07.24
(IT 프로그래머국비지원무료교육/ 무료취업교육)자바,C#,모바일,IoT,UI/UX,웹퍼블리싱,정보처리기사,아두이노 무료교육	02.07

질문/답변

+

질문ecommerce최종본에 관한 질문입니다.	1
답변spring data jpa 질문입니다.~ 참고하세요	1
질문spring data jpa 질문입니다.	0
질문질문이요	1
질문taglib에서 에러가 납니다.	4

State

 현재 접속자	111 명
 오늘 방문자	1,082 명
 어제 방문자	1,470 명
 최대 방문자	3,092 명
 전체 방문자	5,367,576 명
 전체 게시물	21,037 개
 전체 댓글수	7,611 개
 전체 회원수	21,074 명

- 이용약관
- 개인정보취급방침
- 문의하기

- 모바일버전

오라클자바커뮤니티 © All Rights Reserved.

아이디

비밀번호

Login

☐ Remember me

Member

- 회원가입
- 정보찾기

Search

- 자주하는 질문
- 게시글 검색

Misc

- 새글모음
- 현재접속자

Search

포스트 ▼

검색어

To Top