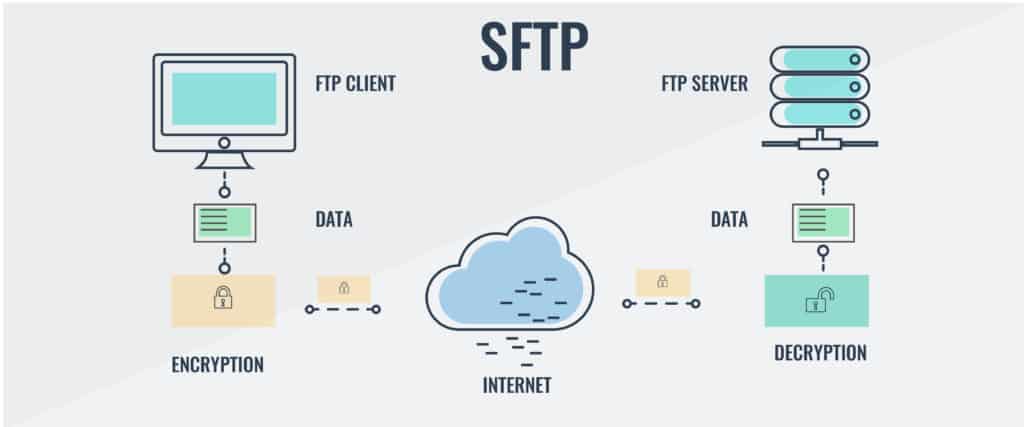
*Publicando arquivo texto em servidor remoto através de código Java usando Protocolo SFTP.*

O que significa SFTP?

SFTP significa “**SSH File Transfer Protocol**” (protocolo de transferência de arquivos SSH) ou , “Secure File Transfer Protocol”, (protocolo seguro de transferência de arquivos) . Com o uso do SFTP, a sua transferência de arquivos será realizada de maneira muito mais segura do que o FTP. A segurança é similar à de uma conexão SSH.

Como funciona o SFTP?

A transferência de arquivos dá-se entre um computador chamado "cliente" (aquele que solicita a conexão para a transferência de dados) e um servidor (aquele que recebe a solicitação de transferência). O utilizador, através de software específico, pode selecionar quais arquivos enviar ou receber do servidor. Para existir uma conexão ao servidor, caso o servidor exija, o utilizador informa um nome de utilizador (**ou username, em inglês**) e uma senha **password**, bem como o nome correto do servidor ou seu endereço IP. Se os dados foram informados corretamente, a conexão pode ser estabelecida.



Agora como sabemos o significado e como funciona o protocolo FTP, vamos ao código para fazer essa transferência de um sistema local para um servidor remoto usando código Java.

// Abaixo estamos importando uma biblioteca para auxiliar no processo de FTP  
// URL do site para download da biblioteca: <https://www.chilkatsoft.com/java.asp>

package br.com.ftp;  
import com.chilkatsoft.\*;

public class EscreverEnviarArquivoSFTP {

// Para resolver essa falha siga os seguintes passos ->  
// 1- Clique no seu projeto com botão direito do mouse, vá em BUILD PATH.  
// 2- Vá em configure BUILD PATH.  
// 3- Nas opção Libraries, clique em add external JARs.  
// 4- Adicione o JAR que você baixou do chilkat.  
//5- Clique no JAR adicionado, vá em Native library location e coloque o caminho do seu JAR chilkat.

static {  
 try{  
 System.loadLibrary("chilkat");

} catch (UnsatisfiedLinkError erro) {

System.err.println("Falha ao carregar biblioteca de código nativo.\n" + erro);

System.exit(1)

}

}

public static void main(String argv[]){

//Nesse ponto estamos criando o objeto para realizar a conexão

CkSFtp sftp = new CkSFtp();

// Aqui teremos os tempos de time-out da conexão em milissegundos:

//Defina o tempo limite.

sftp.put\_ConnectTimeoutMs(5000);

sftp.put\_IdleTimeoutMs(15000);

// Conecte-se ao servidor SSH.

// A porta SSH padrão = 22

// O nome do host pode ser um nome de host ou endereço IP.

String hostName = "sftp.exemplo.com.br";

int port = 22;

boolean success = sftp.Connect(hostname,port);

if (success != true) {

System.out.println(sftp.lastErrorText());

return;

}

// Autentique com o servidor SSH. Chilkat SFTP suporta

// tanto a autenticação baseada em senha quanto a autenticação de chave pública

// Este exemplo usa autenticação por senha.

success = sftp.AuthenticatePw("Login","Password");

if (success != true) {

System.out.println(sftp.lastErrorText());

return;

}

// Após a autenticação, o subsistema SFTP deve ser inicializado

success = sftp.InitializeSftp();

if (success != true) {

System.out.println(sftp.lastErrorText());

return;

}

// Abra um arquivo no servidor para gravação.

// "createTruncate" significa que um novo arquivo é criado; se o arquivo já existir, ele será aberto e truncado.

// Para enviar o arquivo para um Diretório especifico: "/DEV/IN/Arquivo.txt"

String handle = sftp.openFile("Arquivo.txt","escrever","createTruncate");

if (sftp.get\_LastMethodSuccess() != true) {

System.out.println(sftp.lastErrorText());

return;

}

// Escreva o conteúdo texto no arquivo:

success = sftp.WriteFileText(handle,"ansi","Teste Escrita Arquivo - ");

if (success != true) {

System.out.println(sftp.lastErrorText());

return;

}

success = sftp.WriteFileText(handle,"ansi","- 1234567890 ");

if (success != true) {

System.out.println(sftp.lastErrorText());

return;

}

success = sftp.WriteFileText(handle,"ansi"," - ABCDEFGHIJKLMNOPQRSTUVWXYZ");

if (success != true) {

System.out.println(sftp.lastErrorText());

return;  
}

// Feche o arquivo.

success = sftp.CloseHandle(handle);

if (success != true) {

System.out.println(sftp.lastErrorText());

return;

}

System.out.println("Success.");

}

}

Bibliografia.

* <https://pt.wikipedia.org/wiki/File_Transfer_Protocol>
* <https://www.chilkatsoft.com/java.asp>