

Técnicas de programación segura: HTTPS

Explicación

Este código realiza una solicitud HTTPS a una URL específica y muestra la respuesta del servidor.

Se define una URL por parámetros de línea de comandos que representa el servidor al que se realizará la solicitud HTTPS.

- Se crea una instancia de URL con la URL proporcionada.
- Se abre una conexión HTTPS (HttpsURLConnection) con el servidor.
- Se configura la conexión para realizar una solicitud GET mediante el método `setRequestMethod("GET")`.
- Se lee la respuesta del servidor línea por línea utilizando un `BufferedReader`.
- La respuesta se concatena en un `StringBuilder`.
- Por último, se muestra la respuesta del servidor con la salida estándar.

Obviamente nos devolverá texto plano, pero luego se podrá utilizar para recoger datos de la web o tratarlos individualmente e incluso añadir la información a nuestra base de datos.

Este código puede ser útil para realizar solicitudes HTTPS en aplicaciones Java y manejar las respuestas del servidor de manera adecuada. En una aplicación real, deberíamos manejar posibles excepciones y errores de la manera apropiada, pero en este programa al ser un simple ejemplo vamos a lanzar la excepción sin más.

Además, para usar `HttpsURLConnection`, hay que asegurarse de tener los certificados SSL necesarios configurados en nuestro entorno.

Código fuente

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import javax.net.ssl.HttpURLConnection;
/**
 * Esta clase realiza una solicitud HTTPS a un servidor y muestra la respuesta.
 */
public class HttpsExample {
    /**
     * Método principal que realiza la solicitud HTTPS y muestra la respuesta.
     * @param args Argumentos de línea de comandos. Se espera que el primer argumento sea
la URL a la que se realizará la solicitud.
     * @throws IOException Si hay algún error de entrada/salida durante la conexión
     */
    public static void main(String[] args){
        // Verificar si se proporcionó la URL como argumento
        if (args.length < 1) {
            System.out.println("Debe proporcionar la URL como argumento.");
            return;
        }
        try {
            // Obtener la URL del primer argumento
            String httpsURL = args[0];

            // Crear una conexión HTTPS
            URL url = new URL(httpsURL);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();

            // Configurar la conexión
            conn.setRequestMethod("GET");

            // Obtener la respuesta del servidor
            BufferedReader in = new BufferedReader(new InputStreamReader(
                                                                    conn.getInputStream()
                                                                    ));

            String inputLine;
            StringBuilder response = new StringBuilder();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();

            // Mostrar la respuesta del servidor
            System.out.println("Respuesta del servidor:");
            System.out.println(response);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Bibliotecas y dependencias

El programa de ejemplo propuesto no utiliza ninguna dependencia externa aparte de las clases estándar de Java. Utiliza las clases proporcionadas por el paquete `java.net` y `javax.net.ssl` para realizar la conexión HTTPS y manejar la entrada/salida de datos.

`java.io.BufferedReader`: Se utiliza para leer la respuesta del servidor línea por línea.

`java.io.IOException`: Se utiliza para manejar excepciones relacionadas con entrada/salida.

`java.io.InputStreamReader`: Se utiliza para leer la respuesta del servidor.

`java.net.URL`: Se utiliza para representar la URL a la que se realizará la solicitud.

`javax.net.ssl.HttpURLConnection`: Se utiliza para realizar la conexión HTTPS con el servidor.