JAVA Studio Server
SSLSettingAPI

The purpose of this document is to introduce the SSLSettingAPI of Studio JAVA Server(v5.1.3.0243).

This document is used as an internal design record. No information should be exposed to outside of LANDesk.

Revision History

Rev.	Date	Author	Reason for Changes
.01	22.9.2016	Cindy	Initial document
.02	11.10.2016	Cindy	Add API SetSSLSettingForStandaloneAPP Update SSLSettingsParam methods
.03			
.04			
.05			
.05			

API -- SetSSLSettingForStandaloneApp

Syntax

Java

```
public static synchronized final void
SetSSLSettingForStandloneApp(
    SSLSettingsParam param
)
```

Parameters

```
param
    Type:SSLSettingsParam
```

The SSL Settings.

Remarks

The **SetSSLSettingForStandaloneApp** function attempts to set SSL settings in Java application.

Requirements

Import	import com.wavelink.studio.api.SSLSettingAPI; import com.wavelink.studio.api.SSLSettingsParam;
Library	Wlstudio.jar Wavelink.jar

SSLSettingsParam class

Represents the SSL setting property.

Syntax

Java

```
public class SSLSettingsParam
```

Members

Methods

Method	Description
`public SSLSettingsParam()`	Use to initialize an instance of SSLSettingsParam.
`public SSLSettingsParam(boolean isUseSSL)`	Use to initialize an instance of SSLSettingsParam. **@param** isUseSSL Used to set the use SSL flag(true/false). To indicate if this SSL setting will be applied.
`public String getContextProtocol()`	Get the protocol of SSLContext. **@return** The protocol string.
`public void setContextProtocol(String contextProtocol)`	Set the protocol for initializing the SSLContext. Default value is TLSv1.2. **@param** contextProtocol The protocol string. Null will be ignored.
`public String getKeystorePath()`	Get the full path of the key store. **@return** The full path of the key store.
`public void setKeystorePath(String keystorePath)`	Set the full path of the key store which used to create SSL connections.

	@param keystorePath The full path of the key store.
`public String getKeystoreType()`	Get the type of the key store. **@return** The type of the key store.
`public void setKeystoreType(String keystoreType)`	Set the type parameter of the key store. Default value is PKCS12. **@param** keystoreType The type of the key store.
`public String getKeystorePassword()`	Get the password of the key store. **@return** The password of the key store.
`public void setKeystorePassword(char[] keystorePassword)`	Set the password of the key store. **@param** keystorePassword The password of the key store.
`public String getKeyPassword()`	Get the password of the key. **@return** The password of the key.
`public void setKeyPassword(char[] keyPassword)`	Set the password of the key which you want to use. If it is null, Studio Server will set it as the same as the password of key store. **@param** keyPassword The password of the key.
`public String getKeyAlias()`	Get the alias of the key. **@return** The alias of the key.

`public void setKeyAlias(String keyAlias)`	Set the alias of the key. **@param** keyAlias The alias of the key.
`public String getKeyAlgorithm()`	Get the algorithm of the key manager. **@return** The algorithm of the key manager.
`public void setKeyAlgorithm(String keyAlgorithm)`	Set the algorithm of the key manager. **@param** keyAlgorithm The algorithm.
`public String getTrustKeyStorePath()`	Set full path of the trust key store. **@return** The full path of the trust key store.
`public void setTrustKeyStorePath(String trustKeyStorePath)`	Set the full path of the trust key store which used to create SSL connections. If it is null, Studio Server will use the system property javax.net.ssl.trustStore **@param** trustKeyStorePath The full path of the trust key store.
`public String getTrustKeyStoreType()`	Set type of the trust key store. **@return** The type of the trust key store.
`public void setTrustKeyStoreType(String trustKeyStoreType)`	Set the type parameter of the trust key store. Default value is PKCS12. **@param** trustKeyStoreType The type of the trust key store.
`public String getTrustKeyStorePassword()`	Get the password of the trust key store.

	@return The password of the trust key store.
`public void setTrustKeyStorePassword(char[] trustKeyStorePassword)`	Set the password of the trust key store. If it is null, Studio Server will use the system property javax.net.ssl.trustStorePassword **@param** trustKeyStorePassword The password of the trust key store.
`public String getTrustManagerAlgorithm()`	Set algorithm of the trust key manager. **@return** The algorithm of the trust key manager.
`public void setTrustManagerAlgorithm(String trustManagerAlgorithm)`	Set the algorithm of the trust key manager. **@param** trustManagerAlgorithm The algorithm.
`public boolean isUseSSL()`	Get use SSL flag. **@return** true/false.
`public void setUseSSL(boolean useSSL)`	Set use SSL flag. Use to indicate if this SSL setting will be applied. **@param** useSSL true/false.

Examples

Set SSL settings in Java application

```
package wlapps;

import java.io.FileInputStream;
import java.io.IOException;
```

```java
import java.io.InputStream;
import java.util.Properties;

import com.wavelink.clientui.*;
import com.wavelink.studio.api.SSLSettingAPI;
import com.wavelink.studio.api.SSLSettingsParam;

public class HelloWorld extends WaveLinkApplication {

  public void WaveLinkMain (String [] args) {
    WaveLinkIO ioIface = new WaveLinkIO ();

    try {
      ioIface.RFPrint(0, 0, "      WaveLink      ",
WaveLinkIO.WLCLEAR | WaveLinkIO.WLREVERSE);
      ioIface.RFPrint(0, 1, "  Java Application  ",
WaveLinkIO.WLNORMAL);
      ioIface.RFPrint(0, 3, "    Hello World!    ",
WaveLinkIO.WLNORMAL);
      ioIface.RFPrint(0, 14, "Hit any key to exit ",
WaveLinkIO.WLREVERSE);
      ioIface.GetEvent();
    } catch (WaveLinkError wlErr) {
      //Handle error
    }
  }


  public static void main (String[] args) {
    HelloWorld hw = new HelloWorld ();

    boolean result = false;
    boolean useSSL = true;
    String sslContextProtocol = "TLSv1.2";
    String sslKeystorePath = "C:\\keystore.crt";
    String sslKeystorePassword = "12345678";
    String sslKeyPassword = null;
    String sslKeystoreType = "PKCS12";

    SSLSettingsParam param = new SSLSettingsParam();
       param.setUseSSL(useSSL);
       param.setContextProtocol(sslContextProtocol);
       param.setKeystorePath(sslKeystorePath);

   param.setKeystorePassword(sslKeystorePassword.toCharArray()
);     param.setKeyPassword(sslKeyPassword.toCharArray());
       param.setKeystoreType(sslKeystoreType);
```

```
        SSLSettingAPI.setSSLSettingForStandaloneApp(param);

        hw.startServer(args);
    }
}
```

Q&A