Secure XMLLCE Communication

To make the communication between your client terminals and the Futuro server more secure, you can configure secure XMLLCE communication. XMLLCE refers to Futuro's XML Listening and Communication Engine, which facilitates communication between terminals and the server. When secure XMLLCE communication is enabled, the server and terminals will use JWT tokens to authenticate their communication.

When you start up a terminal, the terminal sends a configuration request to the Futuro server. The Futuro server responds with the information the terminal needs to function properly and communicate with the Futuro server. If secure XMLLCE communication is used, the Futuro server also sends a token and a unique identifier to the terminal.

The terminal will then begin sending requests to the server – to log in, start/stop events, or view kiosk events. If secure XMLLCE communication is enabled, each request will also include the token and the identifier that the terminal received from the server at startup. This token will be signed by a keystore configured in an Interface Host. If the token is valid, the Futuro server will respond to each request with a success or error message.

Terminal Requirements

Configuration Steps:

Configure the Keystore

Configure the Interface Host

Configure the Application Setting

InTouch Terminals and Futuro

Timeclock

Web Time Clock

Interface Host

Server Settings

Terminal Requirements

Secure XMLLCE communication require Futuro version 1.16 or later and one of the following terminals:

FuturoTek terminal version H4 with firmware version 3.4.1 or later

Futuro XML Client version 4.0 or later

Futuro Clock (configured with HTTPS)

FutureTime

This feature cannot be used with Terminal Types that are deprecated/unsupported. If you are using one of these legacy terminals, you must disable the secure_receiver_name setting (see below) or the terminal will not connect with the Futuro server.

Configure the Keystore

You will need to obtain a keystore that contains a public and private key pair. The Futuro server will send the public key to a terminal during the configuration request. The server will use the private key to validate requests from the terminal.

When you save the keystore, take note of its alias and password. You will need to add these to your Interface Host configuration.

The keystore file needs to be placed in the \certificate folder in your FUTURO_HOME folder.

The name of the keystore file and the name of its alias will be configured in your Interface Host parameters.

Configure the Interface Host

Make a copy of the TERMINAL_SERVER Interface Host and save it with a new name.

Make sure the Connection Type is set to HTTP and the Host Type is Receiver.

In the Password field, enter the password of your keystore file.

In the Host Parameters tab, add the KEYSTORE_ALIAS and KEYSTORE_NAME parameters. The KEYSTORE_NAME will be the name of your keystore file. The KEYSTORE_ALIAS will be the alias of the keystore.

You can also add the following parameters to your Interface Host:

TOKEN_AUTO_RENEW is used to automatically renew the token that is used to secure communications between client terminals and the Futuro server. If the TOKEN_AUTO_RENEW Host Parameter is set to TRUE, the server will automatically generate a new token when the existing token expires. The default value for this setting is FALSE, meaning you will have to reload the terminal configuration to generate a new token when the existing token expires.

TOKEN_EXPIRATION_ALLOWED_SKEW_SECONDS allows you to extend the expiration timestamp of the token by a few seconds.

TOKEN_SESSION_TIMEOUT_IN_MINUTES defines how long a token will be valid once Futuro issues it. You can use this parameter to change the token session timeout from its default value of 60 minutes.

Configure the secure_receiver_name Setting

The secure_receiver_name Application Setting is used to enable secure XMLLCE communications between terminals and the Futuro server.

Set the secure_receiver_name setting to the name of the Interface Host you configured above. Available options are Interface Hosts with the Host Type of *Receiver* and the Connection Type of *HTTP*.

In the Application form, you can define the secure_receiver_name for a specific server using the Servers tab, or define it for all servers using the Application Settings tab.