Enabling secure connection between client and server

On this page:

Generating a key pair using Java Keytool

To enable a secure connection between the client and the server, you need to generate two keystore files, each in their own keystores:

- **KeyStore.jks** (contains key and certificate) – server-side – upload it using Teamwork Cloud Admin console.
- **cert.jks** (contains public certificate) – client-side – place it in the <*Application folder*>*certs* directory.

To generate a key pair

---

1. Go to your **Java** directory and open the folder named **bin**.
2. Enter *keytool -genkey -alias MyDomain -keyalg RSA -keystore KeyStore.jks -keysize 2048* and press **Enter**.

   You can give any name to *MyDomain* and **KeyStore.jks**.

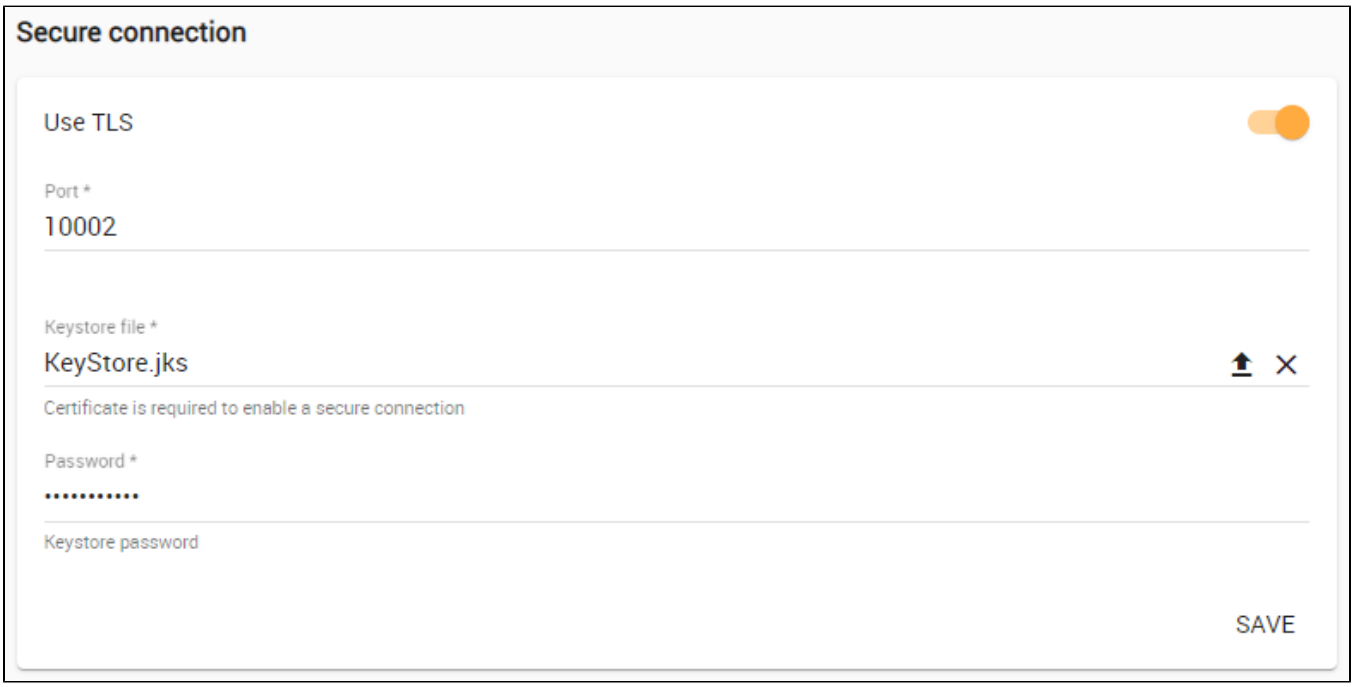
3. Enter a password for your keystore and answer the questions asked. A certificate with a key is generated and placed into the keystore.
4. To extract a certificate without a key, type *keytool -export -alias MyDomain -file certificate.cer -keystore KeyStore.jks* and press **Enter**.
5. To place a certificate into another keystore, e.g. **cert.jks**, type *keytool -import -alias MyDomain -file certificate.cer -keystore cert.jks* and press **Enter**.

   You can give any name to **cert.jks**. However, by default, the tool searches for this exact name.

   Two keystore files are generated: one for the server (**KeyStore.jks**) and one for the client (**cert.jks**). You can now proceed to enable a secure connection between the client (modeling tool) and the server (Teamwork Cloud).

Enabling a secure connection to Teamwork Cloud

To enable a secure connection to Teamwork Cloud, you must enable TLS (Transport Layer Security) on the Teamwork Cloud Admin's **Server Settings** page. This page also allows you to disable the option if you do not need to use a secure connection (see the following figure). You can always enable it whenever necessary.



Enabling TLS protocol in Teamwork Cloud (when the option is turned on, the color changes to orange).

Self-signed TLS certificate warning

Teamwork Cloud Admin uses TLS (Transport Layer Security) as the security protocol to keep any information you enter on Teamwork Cloud Admin private and secure.

By default, your server generates a TLS certificate and signs it as being valid (self-signed certificate). Unlike a TLS certificate issued by a valid Certificate Authority (CA), the self-signed TLS certificate allows a secure connection to be established but does not verify the authenticity of the server.

Trusted root certificates are embedded into popular browsers such as Firefox and Chrome. They are used to verify all TLS certificates that the browsers encounter. If a certificate is not signed by one of these roots, the browsers display an error or warning message stating that it is untrusted. Thus, when you try to access the server via the self-signed one, you will get an error or warning in your web browser. The following figure below shows an example of the TLS certificate untrusted warning in Chrome.

Setting up TLS in Teamwork Cloud Admin

To enable a secure connection using the TLS protocol in Teamwork Cloud Admin

1. Go to the **Settings** app.
2. Move the slider to the right to enable the TLS protocol.
3. Type in the port, upload a **Java Key Store** file, and type the password.
4. Click **Save**.

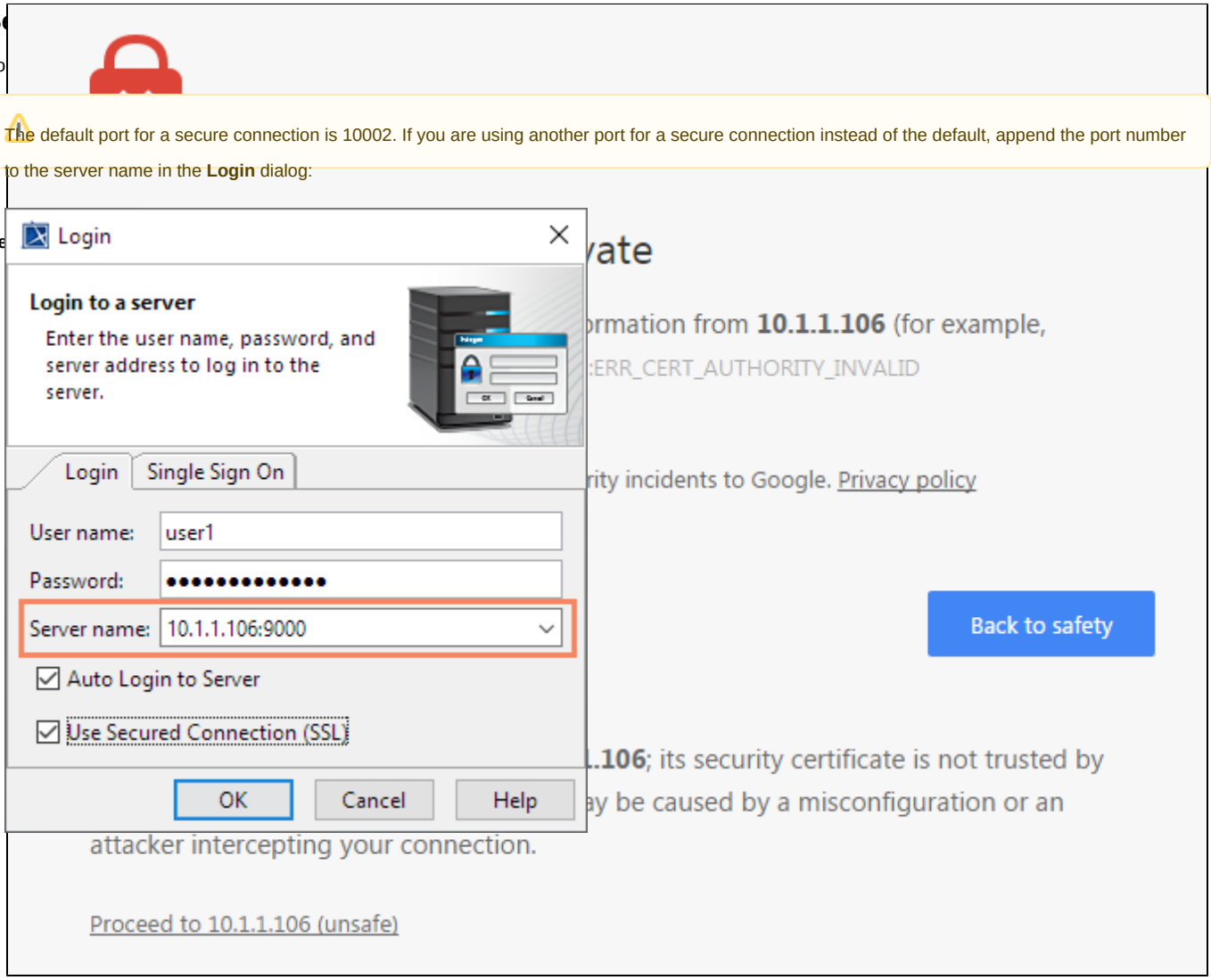
At this point, you will be able to connect to Teamwork Cloud from the modeling tool via the TLS connection.

Se

To

Re

The default port for a secure connection is 10002. If you are using another port for a secure connection instead of the default, append the port number to the server name in the **Login** dialog:



A self-signed TLS certificate error in Google Chrome.

This warning tells you the TLS certificate installed on your server was self-signed and cannot be verified by the browser. You may simply let your browser accept it and continue using the server. If you are using Firefox, you can accept it and the error or warning will no longer appear. If you are using Chrome, the error or warning will appear every time you try to access your server.