



## STAPI User Guide

Document which details the steps of installing and configuring the STAPI client. Included are instructions to follow during installation, troubleshooting and an overview of how your personalized program should work.

**Published: 1 August 2017**

## Table of Contents

|          |                                                 |           |
|----------|-------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction .....</b>                       | <b>3</b>  |
| 1.1      | Required steps .....                            | 3         |
| 1.2      | Features .....                                  | 3         |
| <b>2</b> | <b>Process Overview .....</b>                   | <b>4</b>  |
| 2.1      | Start the client .....                          | 4         |
| 2.2      | The XML String .....                            | 4         |
| 2.3      | Internal Socket .....                           | 5         |
| 2.4      | STAPI to Secure Trading Gateways and back ..... | 6         |
| 2.5      | Other Operations.....                           | 6         |
| <b>3</b> | <b>Pre-Requisites .....</b>                     | <b>7</b>  |
| 3.1      | Your Server .....                               | 7         |
| 3.2      | Internet Merchant Account.....                  | 7         |
| 3.3      | Secure Trading Account.....                     | 7         |
| 3.4      | PCI Accreditation .....                         | 8         |
| <b>4</b> | <b>Installation .....</b>                       | <b>9</b>  |
| 4.1      | Unpacking .....                                 | 9         |
| 4.2      | Certificate/Key generation.....                 | 9         |
| 4.3      | Certificate Import.....                         | 10        |
| 4.4      | Security Considerations .....                   | 11        |
| 4.5      | Configuration .....                             | 12        |
| 4.6      | Port Security.....                              | 14        |
| 4.7      | Proxy (optional) .....                          | 14        |
| 4.8      | Running the STAPI client.....                   | 15        |
| <b>5</b> | <b>XML Code.....</b>                            | <b>16</b> |
| 5.1      | XML Request .....                               | 16        |
| 5.2      | XML Response.....                               | 17        |
| <b>6</b> | <b>Testing.....</b>                             | <b>18</b> |
| 6.1      | Test site reference .....                       | 18        |
| 6.2      | Sending a request.....                          | 18        |
| 6.3      | The response .....                              | 19        |
| <b>7</b> | <b>Troubleshooting.....</b>                     | <b>22</b> |
| 7.1      | Installation problems .....                     | 22        |
| 7.2      | XML Request problems.....                       | 24        |
| <b>8</b> | <b>Going Live.....</b>                          | <b>25</b> |
| 8.1      | Update your XML Requests .....                  | 25        |
| 8.2      | Contact Secure Trading .....                    | 25        |
| 8.3      | Further maintenance .....                       | 25        |
| <b>9</b> | <b>Further Information and Support .....</b>    | <b>26</b> |
| 9.1      | Secure Trading Support.....                     | 26        |
| 9.2      | Secure Trading Sales.....                       | 26        |
| 9.3      | Useful Documents.....                           | 26        |
| 9.4      | Frequently Asked Questions .....                | 26        |

## 1 Introduction

Thank you for choosing the Secure Trading API (STAPI) solution. STAPI is a java application that allows you to process transactions through the Secure Trading Payment Platform (STPP). This is achieved by submitting a series of XML Requests and Responses which will provide you with greater flexibility in terms of developing your system.

The purpose of this document is to provide you with an overview of the installation and configuration of your system and how to begin processing transactions through STPP. It also aims to provide you with a more detailed understanding of using the STAPI system. It details the communications of the STAPI system, and also provides detailed examples and explanations of the XML protocol.



Although the STAPI client itself is a java application, you can use any programming language you wish in order to communicate with the client.

### 1.1 Required steps

Before you can begin processing live transactions using STAPI, please complete the steps listed below. These are to ensure requests are sent securely and reliably:  
(You can begin testing before these steps are completed.)

- Ensure your server is compatible**  
See section 3.1
- Obtain Internet merchant account**  
See section 3.2
- Sign up for Secure Trading account**  
See section 3.3
- Ensure you are PCI accredited when handling card details on your servers**  
See section 3.4
- Install the client on your server**  
See section 4
- Start the STAPI client**  
See section 4.8
- Construct an XML Request**  
See section 5.1
- Listen for an XML Response**  
See section 5.2
- Perform test transactions**  
See section 6
- Go live**  
See section 8

### 1.2 Features

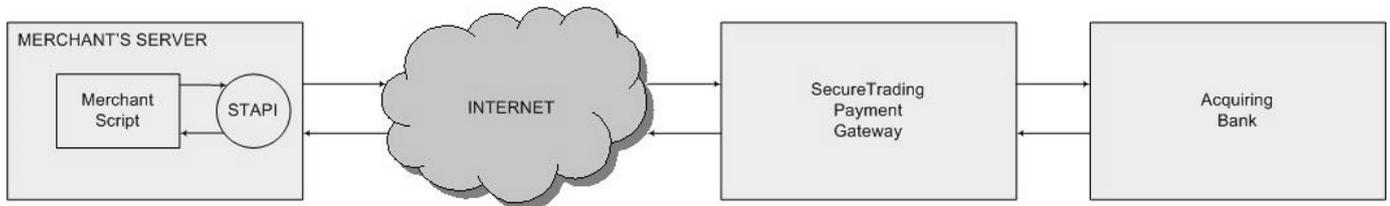
The following features are built into the STAPI client and require no additional configuration:

- // Uses SSL for connections.
- // Checks certificates against up-to-date Certificate Revocation List (CRL).
- // Configured to work with Secure Trading's built-in load-balancer.
- // Implements timeouts handling.

## 2 Process Overview

The following is an overview of how STAPI works from a merchant's perspective, and also how your program should be setup to process transactions via STAPI through STPP.

Below is a diagram that represents the transaction process through STAPI.



There are five key stages of the process:

- # The STAPI client needs to be started on the merchant's server
- # The merchant's system or script will need to generate an XML String that adheres to the ST XML Specification for each request type.
- # The Merchant's system or script will need to send the XML string to the STAPI client via an internal socket.
- # STAPI will then in turn encrypt, send, receive and decrypt the response through the Secure Trading Payment Platform
- # STAPI will return an XML string back to the merchant's system or script.

Each of these stages is explained in greater detail below.

### 2.1 Start the client

The first stage before STAPI can perform transactions is to ensure it is running and that it is ready for a transaction to be submitted. For instructions on installing the client, please refer to section 4 of this document.

### 2.2 The XML String

The XML string that is submitted to the STAPI client will need to adhere to the XML Specification for a given request type, and will include your unique alias and the details of the request you wish to perform.

When submitting a request to Secure Trading, it may be that customer details including their personal information are required. Credit card numbers must be obtained in a **secure** manner, such as the **https** protocol.

Section 3.4 of this document explains the need for PCI accreditation. For this reason Secure Trading strongly recommends that you do not store card details on your server. If you wish to repeat transactions, use the unique transaction reference provided by Secure Trading. For more information, please refer to the **Repeat Payments (Tokenisation)** section of the **XML Specification** document (see section **9.3 Useful Documents**).

## 2.3 Internal Socket

Within your application, you will need to open a TCP socket to pass the XML string through to the STAPI client so it can process your request. This can be achieved in two ways. Please note that as the XML at this point is only being passed internally from your system to the STAPI client, no encryption will be required as the information will not be posted on the internet.

### 2.3.1 Using a standard TCP/IP socket

This is the recommended method used for programming languages which support TCP sockets such as Java, Perl or Python. The XML string is passed from, by default, localhost (127.0.0.1) and through port 5000. If you wish to change the settings for which port your installation of STAPI will be awaiting connections from, please refer to section **4.5 Configuration** on **page 12**.

As an example, below is pseudo-code which instantiates a socket connection to the STAPI client.

```
// Import the socket package
import Socket

// Define the xml string (see The XML string)
xml = "<request><..></..></request>\n"

// Instantiate a new socket with IP Address of 127.0.0.1 and Port 5000
s = new Socket ("127.0.0.1",5000)

// Send the xml down the socket
s.send (xml)

// Receive the response from Xpay
Result=s.receive()

// Close the socket
s.close()
// Result is now available for processing by the merchant
```

Following the request being processed by the STAPI, the data stored in `Result` will be the XML returned.

### 2.3.2 Using the HTTP POST method

This is recommended for languages which do not support TCP sockets directly, such as Cold Fusion. In this case, the XML String is posted to an internal URL.

The equivalent URL that would be entered into an Internet browser would be similar to:

<http://127.0.0.1:5000/?xml=<request><...></..></request>>

In the above example the host address would be 127.0.0.1 and a port of 5000. Inputting the above including the whole XML string that you have generated into a browser is unlikely to be the solution you would prefer. A more suitable way is to program the equivalent in, for example, Cold Fusion:

```
<cfhttp method="post" url=http://127.0.0.1/ port="5000">  
<cfhttpparam encoded="No" TYPE="cgi" NAME="stapi" VALUE="#xml#">  
</cfhttp>
```

This assumes that a variable named "xml" has already been created with the relevant XML string as outlined in the XML Specification for the given request type.

The request should not be URL encoded (as indicated by **encoded="No"** in the above example). Your client code may need to specify the **Content-Type** in the HTTP headers to be **"text/xml"**.



**Please ensure a POST is submitted in production as a GET will not be URL decoded and therefore may not work for certain characters.**

## 2.4 STAPI to Secure Trading Gateways and back

Once your system has submitted the XML string to the STAPI client, then no further action is required until you receive a response. The client will forward the encrypted request to STPP, which in turn will forward the request onto your acquiring bank. They will then authorise or decline the transaction and then send the response to STPP which will be encrypted and sent back to the STAPI client.

## 2.5 Other Operations

Other operations performed through STPP such as refunds, repeats, subscriptions etc all follow the same pattern. Firstly an XML string is generated and sent through a socket to STAPI. This in turn sends the details to STPP which will process the request and return a response to STAPI. Following this, the STAPI will send the response back down the socket to the merchant's script.

### 3 Pre-Requisites

The pre-requisites that you will need to complete in order to begin processing payments through Secure Trading Payment Platform (STPP) are outlined in this section of the document.

#### 3.1 Your Server

You will need to ensure that STAPI is to be installed on a system where:

- # There is a compatible Java environment. This will need to be Java JDK 1.7 or 1.8. To download Java, visit the Oracle website: <http://www.oracle.com/technetwork/java/index.html>
- # You have access to a command prompt or shell with administrator access.
- # There is Internet access from your server.

The recommended hardware requirements are a 600 MHz processor with 256 MB of RAM. Typical platforms that run the API include IBM compatible computers running a Linux/Windows operating system.



You may need to open your firewall for Secure Trading's IP Ranges. Current IP Ranges can be viewed under the "STAPI" column at <http://webapp.securetrading.net/ips.html>

#### 3.2 Internet Merchant Account

An Internet Merchant Account is required if you would like to process online transactions. Secure Trading have relationships in place with certain acquirers and will therefore be able to assist you. For contact details of our sales team, please refer to section 9.2.

#### 3.3 Secure Trading Account

In order to process transactions through Secure Trading's servers, you will need to have an account with us and an alias. You are provided with two Secure Trading aliases when you sign up (one for live transactions and another for test purposes) and these are used to uniquely identify your account when you submit any data to Secure Trading. It should also be quoted in any correspondence with Secure Trading.

Your alias consists of an alphanumeric string (usually including your company name) which is unique to your account. When you reference it, you will need to be aware it is **case sensitive**.



**Please note** that your site reference will be the same as your alias.

You will also be provided with a MyST username and password to allow you to perform certain management tasks on your account.

For more information on becoming a Secure Trading merchant, please contact our Sales team (see section 9.2). If you already have a Secure Trading account, but do not know your site reference, please contact our support team (see section 9.1).



If you do need to contact Secure Trading support, please be aware that for security reasons we may only speak to an authorised contact of the account.

### 3.4 PCI Accreditation

When card details are handled on your servers, you will be required to undertake a form of PCI accreditation, which will differ depending on the size of your company and the volume of transactions that are processed by your system.

For more information on PCI accreditation, please visit:

<https://www.pcisecuritystandards.org>



**Care must be taken to ensure that the STAPI and associated files, including any data requested or returned by it, remain secure on your server. It is recommended that you do not store credit card details on your server.**

## 4 Installation

For the first part of the installation, you will need to ensure that you have the STAPI client. In order to obtain the STAPI client, please download either the Java 1.7 or 1.8 clients using one of the following links.

### Secure Trading API Client (Java 1.7)

[http://webapp.securetrading.net/examples/STAPI\\_JAVA1.7.zip](http://webapp.securetrading.net/examples/STAPI_JAVA1.7.zip)

### Secure Trading API Client (Java 1.8)

[http://webapp.securetrading.net/examples/STAPI\\_JAVA1.8.zip](http://webapp.securetrading.net/examples/STAPI_JAVA1.8.zip)

The examples outlined below are for MS-DOS and Linux or UNIX like platforms, but a similar approach is taken for other operating systems.

### 4.1 Unpacking

You will need to unzip the contents of the stapi.zip into a new directory on your server, for example `c:\stapi` on a Windows box or `/usr/local/stapi` on a Linux or UNIX box.

Most Operating Systems now come with a program to allow you to unzip files. If you don't have a copy you can download WinZip from [www.winzip.com](http://www.winzip.com). Alternatively you could use 7-zip, an open source utility for manipulating archives which can be found at [www.7-zip.org](http://www.7-zip.org).

### 4.2 Certificate/Key generation

Once you have successfully unpacked the contents of the stapi.zip you will need to generate your certificate request to Secure Trading's systems, and also your private keystore.

Before executing the below example commands, you will need to ensure that the version of java you are using has been included within the operating systems "path" environment variable. First find the directory where the Java JDK has been installed, then enter the following commands at the command prompt (the path to your Java installation directory may differ from the example path):

| Windows Example                                                                         | Unix or Linux Example                              |
|-----------------------------------------------------------------------------------------|----------------------------------------------------|
| <pre>PATH=%PATH%;C:\Path-To- Java\jdk1.8.0_60\bin;C:\Pr ogram Files\Java\jre8\bin</pre> | <pre>export PATH=./usr/local/jdk1.8.0_60/bin</pre> |

You will need to ensure that keytool is included within your current path. This should have been done when you setup your path above. To check this has been setup correctly, type `keytool` into the command prompt. This should return an output with a list of commands. If you receive an error or nothing at all check that you have setup your path correctly, ensuring that you referenced your directory for the exact version of java that you downloaded.

For the example below, we will use the alias "site1234"; you should replace this with your alias. Please note that the alias is usually your site reference. You will need to create your own 'good password' (see glossary) whilst following these commands:

| Windows Example                                                                                                    | Unix or Linux Example                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <pre>c: cd stapi java -jar InitStApi.jar Enter Alias: site1234 Enter new password for site1234.jks: pa55Word</pre> | <pre>cd /usr/local/stapi java -jar InitStApi.jar Enter Alias: site1234 Enter new password for site1234.jks: pa55Word</pre> |



The above commands are **case sensitive** and assume that STAPI has been saved in `c:\stapi` on a Windows box, or `/usr/local/stapi` on a Linux or Windows box.

Once the above has been run successfully, you should have two new files within your STAPI directory:

- # **site1234.jks** - This is your private keystore, and must be kept secure as it contains your private key.



**If the file is compromised you should contact Secure Trading immediately and generate a new keystore in order to ensure that you do not lose any confidential information. Do not email this file!**



**Please note** that if you wish to use the STAPI client on another machine, you can copy this file to another machine.

- # **site1234.req.pem** - This is your certificate request.

You will need to send your certificate request (`site1234.req.pem`) to Secure Trading support (see section **9.1 Secure Trading Support** on **page 26**), who will in return generate a certificate for you.

### 4.3 Certificate Import

Following receipt of your new “.pem” file, Secure Trading support will verify your authenticity, then digitally sign your certificate request and return you two new files. These will be:

**site1234.cert.pem**, this is your digitally signed certificate from Secure Trading which will be recognized by our systems as matching your account when you attempt to submit a transaction, identifying you as a genuine merchant.

**merchant256\_ca.cert.pem**, this is the certification authority used.

You will need to place these files within your STAPI directory and then run the following commands:

| Windows Example                                                                                                            | Unix or Linux Example                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <pre>c: cd \stapi java -jar ImportStApi.jar Enter Alias: site1234 Enter keystore password for site1234.jks: pa55Word</pre> | <pre>cd /usr/local/stapi java -jar ImportStApi.jar Enter Alias: site1234 Enter keystore password for site1234.jks: pa55Word</pre> |

This will then attempt to read the files from the current directory and import them into your keystore. You will need to check the output, ensuring it has no errors, and that it has run successfully.

Once this has been completed successfully, you will now have a keystore (site1234.jks) that contains all the necessary certificate chains and authorities. The STAPI client itself is within the file "StApi.jar". All other files ending with ".pem" may now be deleted.

You may also change the name of the keystore (site1234.jks) if you wish, but you will need to ensure that you reference it correctly within the configuration file outlined in section 4.5.



If you are encountering errors when importing your certificate, ensure you are importing both files exactly as they were sent, without any modifications or additional whitespace. Please refer to section 7.1 for further information.

#### 4.4 Security Considerations

When performing STAPI requests a certificate will be used to establish a secure connection to one of the Secure Trading payment servers.

It is important that adequate security measures are adhered to in order to safeguard STAPI certificates and prevent certificates from becoming compromised and used by others for illicit use.

When handling STAPI certificates it is recommended that these guidelines are followed:

- # Physically protect the certificates from unauthorised access
- # Restrict access to the certificates to the fewest number of people necessary
- # Ensure that expired certificates are deleted and not kept on company systems

If at any time you think that the security of your certificates has been compromised please contact the support at Secure Trading immediately. The telephone numbers and email address may be found in section 9.1 **Secure Trading Support** on page 26.

In addition, you will need to ensure that the keystore and its corresponding password are kept securely from unauthorised users in order to ensure the confidentiality of the encrypted information. You should ensure the keystore file and STAPI configuration file are both kept securely and should be accessible only by the STAPI client and authorised maintainers.

## 4.5 Configuration

In order for STAPI to work, an XML configuration file will need to be setup. By default, this file will be called “stapi.ini” and will need to be stored within the STAPI directory.

Please find below an example “stapi.ini” file:

```
<stapi>
<port>5000</port>
<host>127.0.0.1</host>
<loglevel>INFO</loglevel>
<printlevel>INFO</printlevel>
<logfile>stapi.log</logfile>
<responsetype>HTTP/1.1</responsetype>
<keystore alias="site1234">
    <file>/path/to/site1234.jks</file>
    <password>pa55Word</password>
</keystore>
</stapi>
```

- # An example of this configuration file is included in the stapi.zip directory.
- # This is called “example\_merchantstapi.ini”.
- # You will need to update the **<keystore alias>**, **<file>** and **<password>** tags with your unique values, and rename this to be called “stapi.ini”.

When using more than one site, the “stapi.ini” file is configured as follows:

```
<stapi>
<port>5000</port>
<host>127.0.0.1</host>
<loglevel>INFO</loglevel>
<printlevel>INFO</printlevel>
<logfile>stapi.log</logfile>
<responsetype>HTTP/1.1</responsetype>
<keystore alias="site1234">
    <file>/path/to/site1234.jks</file>
    <password>pa55Word</password>
</keystore>
<keystore alias="site5678">
    <file>/path/to/site5678.jks</file>
    <password>pa66W0rd</password>
</keystore>
</stapi>
```



You may reference and call the “stapi.ini” file by another name (see section 4.8 for an example of this).

Please find below an explanation of each field:

| Tag                | Type | Length | Required | Comment                                                                                                                                                                                                                                                                                                                           |
|--------------------|------|--------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stapi              |      |        |          |                                                                                                                                                                                                                                                                                                                                   |
| port               | n    | 255    | N        | The port on which STAPI will listen for new connections. Default is 5000.                                                                                                                                                                                                                                                         |
| host               | an   | 15     | N        | The host on which STAPI will listen for new connections. Default is 127.0.0.1 (localhost).                                                                                                                                                                                                                                        |
| loglevel           | an   | 255    | N        | The amount of information STAPI will store in the logfile and print to screen respectively.<br><br>'ALL' must NOT be used in a production environment since it will cause sensitive information to be logged unencrypted. This can be useful during integration development.<br><br>'FINE' provides more information than 'INFO'. |
| printlevel         | an   | 255    | N        | 'INFO' is best used for production environment.<br><br>'SEVERE' logs a minimal amount of data.                                                                                                                                                                                                                                    |
| logfile            | an   | 255    | N        | URI to a writeable file. Defaults to stapi.log in the current directory. The directory that contains the log file must also be writeable to allow for the creation of a lock file.                                                                                                                                                |
| timeout            | n    | 255    | N        | Number of milliseconds to wait for data on idle socket connections before aborting. The default values should only be changed if you are experiencing problems with connectivity.                                                                                                                                                 |
| sslconnect timeout | n    | 255    | N        |                                                                                                                                                                                                                                                                                                                                   |
| sslreceive timeout | n    | 255    | N        |                                                                                                                                                                                                                                                                                                                                   |
| responsetype       | an   | 255    | N        | This will cause STAPI to return HTTP headers before the XML response which is useful if the merchant wants to use a HTTP client to communicate with STAPI instead of using plain TCP socket connections.                                                                                                                          |
| keystorealias      | an   | 255    | Y        | Your alias. If STAPI is to be used to process transactions through multiple aliases you must provide a separate < > section for each alias.                                                                                                                                                                                       |
| file               | an   | 255    | Y        | URL to the keystore file generated in sections 4.2 and 4.3.                                                                                                                                                                                                                                                                       |
| password           | an   | 255    | Y        | The password for the keystore.                                                                                                                                                                                                                                                                                                    |
| retries            | n    | 255    | N        | The number of attempts STAPI will try to connect to the Secure Trading gateway network before aborting the transaction. To maintain an uninterrupted service this value should be left at the default value.                                                                                                                      |
| system             | an   | 255    | N        | Optional java system properties. Useful examples include http.proxyPort and https.proxyPort. Some JVM's may need to set "http.proxySet" to "true" as well. If you do not use a proxy and have no requirements for java system properties omit the entire <system> section from stapi.ini                                          |



**Please note** that for the XML Requests, the <alias> tags will match the <keystore alias> specified in the configuration file.

#### 4.6 Port Security

STAPI uses a number of ports in order to communicate with STPP and the merchant's system. The ports used by STPP are outlined below.

| Number | Description                                                                                                                                                                                                         |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5000   | This is the default port used to receive incoming requests from your system through an internal socket. Please note that you can change this port to be whichever port you wish by configuring your stapi.ini file. |
| 443    | This port is used by STAPI to send and receive encrypted information to and from the STPP payment gateway network. You will need to ensure that this port is open on your firewall for outgoing connections.        |
| 80     | This port is used to check for any new gateways and certificate revocations.                                                                                                                                        |



**Your STAPI client can be used to view details of transactions processed on your SecureTrading account and authorise new payments. In the interest of security, it is imperative that you ensure that only trusted sources are permitted to access the STAPI client on your system.**

#### 4.7 Proxy (optional)

It is recommended that you configure your STAPI client to connect directly to Secure Trading. However, you can optionally configure the STAPI client to work with a proxy by adding the following to your "stapi.ini" file (see section 4.5 for further information).

```
<system>
<property name="http.proxyHost" value="my.http.proxy.host" />
<property name="https.proxyHost" value="my.ssl.proxy.host" />
</system>
```

You must replace "my.http.proxy.host" and "my.ssl.proxy.host" with the values for your proxy server.

### 4.8 Running the STAPI client

Once you have successfully completed the stages outlined above, you can begin running STAPI with the following commands:

| Windows Example                             | Unix or Linux Example                              |
|---------------------------------------------|----------------------------------------------------|
| <pre>c:\ cd stapi java -jar StApi.jar</pre> | <pre>cd /usr/local/stapi java -jar StApi.jar</pre> |

Or, alternatively, if you have changed the name and location of your configuration file, the last command will be:

| Windows Example                                      | Unix or Linux Example |
|------------------------------------------------------|-----------------------|
| <pre>java -jar StApi.jar /path/to/newstapi.ini</pre> |                       |

Once this has been setup successfully, the command prompt output will look similar to the following:

```
2008-09-11 10:44:47 main Listening on 127.0.0.1:5000
```

This indicates that STAPI is now up and running and is awaiting a connection from your system.

If it has failed to run, please look at the troubleshooting section of this guide (section 7). Alternatively you can contact our support team.

## 5 XML Code

This section explains the basic format of the XML request and responses that are used with the STAPI system. The XML instructs STPP (Secure Trading Payment Platform) to perform certain actions (e.g. seek authorisation for a transaction). Full XML examples can be found in:

- # The [XML Specification document](#) and other XML documentation on Secure Trading's website (see section 9.3).
- # The downloadable directory of XML examples, hosted by Secure Trading. See section 4 for the URLs.

Please note that the XML Tags outlined within this document are all case sensitive. Secure Trading have applied a naming convention to their XML specification which forces all tags to be lowercase with no spaces.

For more information on XML tags, please refer to <http://www.xml.com>

### 5.1 XML Request

All requests sent to Secure Trading originate from one `<requestblock>` block which has one attribute - `version`, which refers to the version of STAPI you are using.

The following is an example of an XML Request for an authorisation, using the site reference "test\_site12345":

```
<requestblock version="3.67">
  <alias>test_site12345</alias>
  <request type="AUTH">
    <operation>
      <sitereference>test_site12345</sitereference>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
    <merchant>
      <orderreference>Example AUTH</orderreference>
      <termurl>https://www.example.com/termurl.cgi</termurl>
      <name>Test Merchant</name>
    </merchant>
    <customer>
      <ip>1.2.3.4</ip>
    </customer>
    <billing>
      <amount currencycode="GBP">2115</amount>
      <town>Bangor</town>
      <country>GB</country>
      <payment type="VISA">
        <expirydate>10/2031</expirydate>
        <pan>4111111111111111</pan>
        <securitycode>123</securitycode>
      </payment>
    </billing>
    <settlement/>
  </request>
</requestblock>
```

## 5.2 XML Response

Secure Trading will respond to all valid requests it receives from merchants. The response will be within the `<responseblock>` tags, and will also have a `version` attribute.

The following is an example of an XML Response sent by Secure Trading:

```
<?xml version='1.0' encoding='utf-8'?>
<responseblock version="3.67">
  <requestreference>X3148177</requestreference>
  <response type="AUTH">
    <merchant>
      <merchantname>Test Merchant</merchantname>
      <orderreference>Example AUTH</orderreference>
      <tid>27880001</tid>
      <merchantnumber>00000000</merchantnumber>
      <merchantcountryiso2a>GB</merchantcountryiso2a>
      <operatorname>test_site12345</operatorname>
    </merchant>
    <transactionreference>4-9-1533116</transactionreference>
    <timestamp>2014-05-20 09:47:17</timestamp>
    <acquirerresponsecode>00</acquirerresponsecode>
    <operation>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
    <settlement>
      <settleduedate>2014-05-20</settleduedate>
      <settlestatus>0</settlestatus>
    </settlement>
    <billing>
      <amount currencycode="GBP">2115</amount>
      <payment type="VISA">
        <issuer>SecureTrading Test Issuer1</issuer>
        <pan>411111#####1111</pan>
        <issuercountry>US</issuercountry>
      </payment>
      <dcc enabled="0"/>
    </billing>
    <authcode>TEST</authcode>
    <live>0</live>
    <error>
      <message>Ok</message>
      <code>0</code>
    </error>
    <security>
      <postcode>0</postcode>
      <securitycode>2</securitycode>
      <address>0</address>
    </security>
  </response>
</responseblock>
```

You will need to analyze the different fields returned in the response to determine whether or not the request was successful (in particular the `<error>` `<code>` and `<settlestatus>` elements). Information on how best to check the XML Response can be found in the Best Practices section of the [XML Specification document](#).

## 6 Testing

### 6.1 Test site reference

You will be provided with a “test” site reference and a “live” site reference on sign up. All test site references are preceded with the word “test\_” (e.g. “test\_merchant12345”). Secure Trading recommends you test your solution thoroughly before processing live transactions. Your test site references will always be available for you to process test transactions, even after going live.

### 6.2 Sending a request

Processing a transaction through STAPI is based on sending an XML string through a socket connection to the STAPI client.

Included in the stapi.zip file are some simple examples of how to write the necessary code to connect to the STAPI client and submit a transaction request. There are examples in different languages, but they all follow a similar process. The client reads in the contents of the “AUTH\_VISA/request.xml” file and sends it through a socket connection to the STAPI client, before finally printing the response from the STAPI client to the screen.

There is a directory for each request type that can be used with STAPI. Within each directory is a request.xml and a response.xml file that demonstrate how the XML Requests and Responses will look for each request type.

It is good practice at this point to attempt to process a transaction on your test site using the examples provided. This will prove that the installation has been successful and that there are no connection issues.

Please note that the included examples contain example data. Some request types such as ‘REFUND’ and ‘ORDERDETAILSAUTH’ have a reference to a parent transaction reference (refer to the **XML Specification** for more information). To test these types of transaction, you will need to process a parent transaction first in order to obtain the ‘transaction reference’ from their response.

To process a test transaction using the Java (“Test.class”) example:

- # Choose a request type to test, for this demonstration we will use the authorisation request.xml (which can be found in the AUTH\_VISA directory).
- # Open the file and insert your alias (provided when you signed up for the account) within the <alias> tags then insert your site reference within the <sitereference> tags, then save the file.
- # Open a Command Prompt or Terminal and type the following commands (The argument specifies the location of the request.xml file to use for the test, in this case “AUTH\_VISA/request.xml”).  
To test a REFUND we would use “REFUND\_VISA/request.xml”)

| Windows Example                                        | Unix or Linux Example                                          |
|--------------------------------------------------------|----------------------------------------------------------------|
| <pre>cd C:\stapi java Test AUTH_VISA\request.xml</pre> | <pre>cd /usr/local/stapi java Test AUTH_VISA\request.xml</pre> |

- # Wait for a response to be shown on screen, this should take somewhere between 2 and 20 seconds depending on your connection speed.

## 6.3 The response

### 6.3.1 Output in the STAPI Client

The output from the example should be similar to the following:

```
C:\stapi>java Test AUTH_VISA\request.xml
Reading from file: AUTH_VISA\request.xml
Please wait while the transaction is authorised on <IP address>
<?xml version='1.0' encoding='utf-8'?>
<requestblock version="3.67">
  <alias>site12345</alias>
  <request type="AUTH">
    <merchant>
      <orderreference>AUTH_VISA</orderreference>
    </merchant>
    <customer>
      <town>Bangor</town>
      <name>
        <middle>Mary</middle>
        <prefix>Miss</prefix>
        <last>Smith</last>
        <first>Joanne</first>
      </name>
      <ip>1.2.3.4</ip>
      <telephone type="H">1111111111</telephone>
      <street>Second Street</street>
      <postcode>CU888ST</postcode>
      <premise>111</premise>
    </customer>
    <billing>
      <town>Bangor</town>
      <name>
        <middle>joe</middle>
        <prefix>Dr</prefix>
        <last>bloggs</last>
        <suffix>Jr.</suffix>
        <first>fred</first>
      </name>
      <country>GB</country>
      <payment type="VISA">
        <expirydate>10/2031</expirydate>
        <pan>4111110000000211</pan>
        <securitycode>123</securitycode>
      </payment>
      <telephone type="M">0777777777</telephone>
      <county>Gwynedd</county>
      <amount currencycode="GBP">100</amount>
      <street>Test Street</street>
      <postcode>TE45 6ST</postcode>
      <premise>789</premise>
      <email>fred.bloggs@example.com</email>
    </billing>
    <operation>
      <sitereference>site12345</sitereference>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
  </request>
</requestblock>
```

```
<?xml version='1.0' encoding='utf-8'?>
<responseblock version="3.67">
  <requestreference>X36964132</requestreference>
  <response type="AUTH">
    <merchant>
      <merchantname>Test Merchant</merchantname>
      <orderreference>AUTH VISA</orderreference>
      <tid>27886000</tid>
      <merchantnumber>00000000</merchantnumber>
      <merchantcountryiso2a>GB</merchantcountryiso2a>
      <operatorname>site12345</operatorname>
    </merchant>
    <transactionreference>23-5-4</transactionreference>
    <timestamp>2014-03-20 09:14:12</timestamp>
    <acquirerresponsecode>00</acquirerresponsecode>
    <operation>
      <accounttypedescription>ECOM</accounttypedescription>
    </operation>
    <settlement>
      <settleduedate>2014-03-20</settleduedate>
      <settlestatus>0</settlestatus>
    </settlement>
    <billing>
      <amount currencycode="GBP">100</amount>
      <payment type="VISA">
        <issuer>Test Issuer</issuer>
        <pan>411111#####0211</pan>
        <issuercountry>ZZ</issuercountry>
      </payment>
      <dcc enabled="0"/>
    </billing>
    <authcode>3</authcode>
    <live>1</live>
    <error>
      <message>Ok</message>
      <code>0</code>
    </error>
    <security>
      <postcode>2</postcode>
      <securitycode>2</securitycode>
      <address>2</address>
    </security>
  </response>
</responseblock>
```

The response from the STAPI client is printed as shown between the `<responseblock></responseblock>` tags. The `<error></error>` tags should contain a tag `<code>0</code>` to signify that the transaction was successful.

If you do not receive an output similar to that above, or you do not receive an output at all please refer to the troubleshooting section.

If you received an output similar to the example above then you have successfully processed a transaction using the STAPI client.

**6.3.2 Output in the Listener**

In the listener, all requests processed are displayed in real time. Each request is shown with the following information:

- # Timestamp the request was processed
- # Unique reference for the request
- # Unique reference for the site the request was processed on
- # Time taken to process the request
- # IP of the gateway that processed the request

The following example illustrates three requests that were successfully processed to STPP.

```
2014-02-20 14:15:47      X493528132      1: User:site12345
      Gateway:https:// 192.168.11.51:443/
2014-02-20 14:15:48      X493528132      Request finished in 0.74 seconds
2014-02-20 14:16:05      X928048771      1: User:site12345
      Gateway:https:// 192.168.11.51:443/
2014-02-20 14:16:05      X928048771      Request finished in 0.08 seconds
2014-02-20 14:16:26      X525914103      1: User:site12345
      Gateway:https:// 192.168.11.51:443/
2014-02-20 14:16:26      X525914103      Request finished in 0.07 seconds
```

## 7 Troubleshooting

### 7.1 Installation problems

If you experience problems installing or starting the STAPI application, please check that:

- # You are using the version of Java specified in the 'Prerequisites' section of this document;
- # Java has been installed correctly and the environment PATH has been set correspondingly;
- # You have exactly one STAPI client running on your server.

Please take care not to 'Copy' and 'Paste' from this PDF when executing commands on the Command Line Interface (CLI); Using your operating systems 'Copy' and 'Paste' functions to copy commands from this PDF document may cause errors when executing those commands on the CLI;. PDF documents may contain hidden characters that are invisible to but may still be executed by the CLI, and cause errors.

The following tables show the cause and solution for common installation issues:

| Error                                                                                                                                             | Cause                                                                                                                                                       | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Exception in thread "main"<br>java.io.IOException: Cannot run program "keytool": CreateProcess error=2, The system cannot find the file specified | The keytool program cannot be found. This is part of the java installation. Make sure that the path has been set properly (refer to section 4.2 on page 9). | The path should include the 'bin' directory of your java installation. This is where your system will look for the keytool program.<br><br>Alternatively you can specify the full path to the keytool to be used e.g.<br>C:\jdk1.8.0_60\bin\keytool<br><br>Your Java jdk may be installed in a different directory to the above example.<br><br>N.B. If you have multiple versions of java installed, ensure only one is specified in the complete path. |
| java.net.BindException: Address already in use                                                                                                    | Another service is running on the same port (default 5000) as STAPI. It could be due to another STAPI client already running.                               | Stop the other service or run STAPI on another port. To use a different port, please see section 4.5 Configuration on page 12.                                                                                                                                                                                                                                                                                                                           |
| keytool error:<br>java.lang.Exception: Certificate not imported, alias <merchant256_ca> already exists                                            | The certificate authority 'merchant256_ca' has already been added to the keystore, perhaps from an earlier attempt to import the certificates               | Remove the files created during the certificate signing process (site1234.req.pem, site1234.jks, site1234.cert.pem and merchant256_ca.cert.pem) then follow section 4.2 Certificate/Key generation on page 9.                                                                                                                                                                                                                                            |
| java.io.FileNotFoundException<br>Exception: stapi.log (Access is denied)                                                                          | The log file defined in stapi.ini for example 'stapi.log' has read only permission.                                                                         | Make sure that the log file defined in stapi.ini for example 'stapi.log' has write access.                                                                                                                                                                                                                                                                                                                                                               |
| java.io.IOException: Couldn't get lock for stapi.log                                                                                              | The directory containing the log file defined in stapi.ini for example 'stapi.log' has read only permission.                                                | Make sure that the directory containing the log file defined in stapi.ini for example 'stapi.log' has write access.                                                                                                                                                                                                                                                                                                                                      |
| Exception in thread "main"                                                                                                                        | If you have updated the port tags within your stapi.ini                                                                                                     | Open your Test.java file, update the line "int port = 5000;" to be the same                                                                                                                                                                                                                                                                                                                                                                              |

| Error                                                                                                                                                                | Cause                                                                                       | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| java.net.ConnectException: Connection refused: connect                                                                                                               | configuration file, but have not updated your Test.java file.                               | port you have specified in your stapi.ini configuration file.<br>Then compile the file by running "javac Test.java" on the command line.<br>Test.java has -h and -p command line arguments for host/port which can be used instead of recompiling Test.java                                                                                                                                                                                                                                                                                         |
| com.securetrading.stapi.StApiException: Unable to verify against Certificate Revocation List                                                                         | This is caused by the relevant ports not being open.                                        | Ensure all ports are open so STAPI can run, please see section <b>4.6 Port Security</b> on page 14.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| An error about "Invalid Status Line" or similar, depending on the language used.                                                                                     | This is caused because STAPI hasn't been configured to return a HTTP response code.         | Update the stapi.ini file to include the missing http header configuration:<br><code>&lt;responsetype&gt;<br/>HTTP/1.1<br/>&lt;/responsetype&gt;</code> .<br>Please see section <b>4.5 Configuration</b> on page 12:                                                                                                                                                                                                                                                                                                                                |
| keytool error:<br>java.security.cert.CertificateException:<br>sun.security.pkcs.ParsingException:<br>ObjectIdentifier() -<br>- data isn't an<br>object ID (tag = 48) | This occurs when the certificate is either incorrect or corrupt.                            | Either re-import the certificate in case of corruption, or re-request the certificate in case it's incorrect.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| keytool error:<br>java.security.cert.CertificateException:<br>java.io.IOException:<br>DerInputStream.getLength():<br>lengthTag=127, too<br>big.                      | This can occur when importing certificates that have been modified.                         | Ensure you are importing both files exactly as they were sent, without any modifications or additional whitespace.<br>Tips:<br># When transferring the files to your server, store them in a zip folder, transfer the entire folder and extract on the target computer. This should prevent the contents of the files from being manipulated while being transferred.<br># If using FTP, try setting the transfer mode to binary instead of ASCII, or vice-versa.<br># If using FTP, try turning on/off passive mode before transferring the files. |
| java.lang.IllegalArgumentException: Bad level "<loglevel>"                                                                                                           | This occurs when the loglevel in the stapi.ini file has been set with an unsupported value. | Change loglevel to either "ALL", "FINE", "INFO" or "SEVERE".<br>See section 4.5 for further information.                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## 7.2 XML Request problems

Check the log file created by STAPI (default is stapi.log) for more detailed error information. You may get more information by changing the loglevel setting in the stapi.ini configuration file (see section **4.5 Configuration** on **page 12**).

The following tables show the cause and solution for common error messages:

| Error                                                                                                                               | Cause                                                                                                                                                         | Solution                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Request aborted:<br>Unable to send to gateway<br>javax.net.ssl.SSLHandshakeException:<br>Received fatal alert:<br>handshake_failure | This may indicate there is a problem with the certificates.                                                                                                   | Ensure that the certificates you have received from Secure Trading have not been corrupted in any way (for example accidentally editing characters in the file). Try following the Import phase again (section <b>4.3 Certificate Import</b> on <b>page 10</b> ). You may need to restart the StApi Client to pick up any changes made. |
| Invalid or incomplete XML<br>java.net.SocketTimeoutException: Read timed out                                                        | This indicates a problem you're your XML request. Either it is not well formed or the complete request is not being received by STAPI.                        | Ensure there is a newline at the end of the XML request. STAPI will only read complete lines.<br><br>Ensure all your XML tags are well formed (each tag must be closed and nested tags must be fully enclosed within other tags).<br><br>Ensure any XML characters are correctly encoded (e.g. '<' must be encoded to '&lt;')           |
| No certificates found for: site1234                                                                                                 | The XML request contains the alias site1234 but either the stapi.ini or the corresponding keystore does not contain the correct configuration for this alias. | Ensure stapi.ini has a <keystore> section with the alias attribute set to site1234. Also ensure the <file> within <keystore> refers to the correct keystore for this alias.                                                                                                                                                             |

## 8 Going Live



Please ensure you have completed all of the steps listed in section 1.1 before going live.

### 8.1 Update your XML Requests

You will need to ensure the XML Requests are using your live site reference in the `<sitereference>` field.

### 8.2 Contact Secure Trading

Once you have tested your system and you are ready to go live, please send an email to [support@securetrading.com](mailto:support@securetrading.com) with your site reference and request to go live. You will receive a response when your live site is ready to begin processing payments.

### 8.3 Further maintenance

The most recent version of the client will run background processes to ensure the following:

- // DNS lookups to the STAPI domains are up-to-date.
- // Check the CRL (Certificate Revocation List) is up-to-date.

The client will regularly check for updates automatically. You do not need to take any action.



The aforementioned background processes run on version 1.13 of the client and newer. You can view the version number of the installed client by running the following at the command prompt: `java -jar StApi.jar --version`

## 9 Further Information and Support

This section provides useful information with regards to documentation and support for your Secure Trading solution.

### 9.1 Secure Trading Support

If you have any questions regarding integration or maintenance of the system, please contact our support team using one of the following methods.

| Method    | Details                                                                                                           |
|-----------|-------------------------------------------------------------------------------------------------------------------|
| Telephone | +44 (0) 1248 672 050                                                                                              |
| Fax       | +44 (0) 1248 672 099                                                                                              |
| Email     | <a href="mailto:support@securetrading.com">support@securetrading.com</a>                                          |
| Website   | <a href="http://www.securetrading.com/support/support.html">http://www.securetrading.com/support/support.html</a> |

### 9.2 Secure Trading Sales

If you do not have an account with Secure Trading, please contact our Sales team and they will inform you of the benefits of a Secure Trading account.

| Method            | Details                                                                 |
|-------------------|-------------------------------------------------------------------------|
| Telephone         | 0800 028 9151                                                           |
| Telephone (Int'l) | +44 (0) 1248 672 070                                                    |
| Fax               | +44 (0) 1248 672 079                                                    |
| Email             | <a href="mailto:sales@securetrading.com">sales@securetrading.com</a>    |
| Website           | <a href="http://www.securetrading.com">http://www.securetrading.com</a> |

### 9.3 Useful Documents

Any document regarding the STPP system can be found on SecureTrading's website (<http://www.securetrading.com>). Alternatively, please contact our support team as outlined above.

### 9.4 Frequently Asked Questions

Please visit the FAQ section on our website (<http://www.securetrading.com/support/faq>).