rws

Installation and Release Notes

XPP RESTful Web Services 1.8

March 2026

Legal notice

Copyright and trademark information relating to this product release.

This product may include open source or similar third-party software, details of which can be found by clicking the following link: null.

Although RWS takes all reasonable measures to provide accurate and comprehensive information about the product, this information is provided as-is and all warranties, conditions or other terms concerning the documentation whether express or implied by statute, common law or otherwise (including those relating to satisfactory quality and fitness for purposes) are excluded to the extent permitted by law.

To the maximum extent permitted by law, RWS shall not be liable in contract, tort (including negligence or breach of statutory duty) or otherwise for any loss, injury, claim liability or damage of any kind or arising out of, or in connection with, the use or performance of the Software Documentation even if such losses and/or damages were foreseen, foreseeable or known, for: (a) loss of, damage to or corruption of data, (b) economic loss, (c) loss of actual or anticipated profits, (d) loss of business revenue, (e) loss of anticipated savings, (f) loss of business, (g) loss of opportunity, (h) loss of goodwill, or (i) any indirect, special, incidental or consequential loss or damage howsoever caused.

All Third Party Software is licensed "as is." RWS makes no warranties, express, implied, statutory or otherwise with respect to the Third Party Software, and expressly disclaims all implied warranties of non-infringement, merchantability and fitness for a particular purpose. **In no event will Licensor be liable for any damages, including loss of data, lost profits, cost of cover or other special, incidental, consequential, direct, actual, general or indirect damages arising from the use of the Third Party Software or accompanying materials, however caused and on any theory of liability. This limitation will apply even if Licensor has been advised of the possibility of such damage. The parties acknowledge that this is a reasonable allocation of risk.**

Contents

1

Release Notes

Version 1.8.0

Release Date: March 2026

A significant new security feature has been added for securing access to XPP REST: X.509 authentication with client certificates. The NodeJS runtime has also been updated to v22.22.

Fixed Issues

Issue #	Description
XPP-12667	Enhancement: client certificate authentication
CRQ-43426	Bug: Installation log location not appearing
XPP-12981	Vulnerability: minimist:0.2.4
XPP-12704	Vulnerability: tmp:0.2.3
XPP-12961	Vulnerability: node:22.19.0
XPP-12705	Vulnerability: form-data:4.0.1
XPP-12980	Vulnerability: diff:7.0.0

Version 1.7.0

Release Date: August 2025

NodeJS has been updated on Windows to 22.17.1. Additional weak TLS cipher suites have been disabled.

Fixed Issues

Issue #	Description
CRQ-42888	Vulnerability: weak TLS cipher suites
CRQ-43160	Vulnerability: Node.js (Windows)

Version 1.6.0

Release Date: July 2025

Various dependencies have been upgraded to address vulnerabilities reported against them. NodeJS has been updated to version 22.17. Weak TLS cipher suites have also been disabled.

Fixed Issues

Issue #	Description
XPP-12030	Enhancement: offline documentation support
CRQ-42888	Vulnerability: weak TLS cipher suites
CRQ-42834	Vulnerability: Node v22.11
XPP-12391	Vulnerability: formidable
XPP-12096	Vulnerability: serialize-javascript
XPP-12094	Vulnerability: swagger-ui-bundle.js
XPP-12397	Bug: OpenAPI documentation contains unsupported option value

Version 1.5.0

Release Date: March 2025

It is now easier to understand and implement XPP RESTful Web Services because its online documentation has been migrated to Swagger/OpenAPI. Two new HTTP response headers have been added: (1) *XYV_ADVP_NODE* identifies the XPP server node that XPP REST uses, and (2) `Strict-Transport-Security` is an implementation of HSTS. Additionally, security Upgrades have been made to some third-party components, including NodeJS, which has been updated to version`22.11.0`.

Fixed Issues

Issue #	Description
CRQ-40820	Enable HTTP Strict Transport Security (HSTS) Headers
XPP-10783	Vulnerability: `bootstrap.min.js`
XPP-10977	Vulnerability: find-my-way v7.7.0
XPP-10967	Vulnerability: send v0.18.0
XPP-11242	Vulnerability: foreground-child:3.2.1/cross-spawn^7.0.0
XPP-11152	Migrate Documentation to OpenAPI and Swagger
XPP-11346	Add ADVP Node Name to HTTP Response Headers
XPP-10540	Silent Install (`-i silent`) does not configure installation with default port number `2995` resulting in REST service failure
XPP-9675	Replace deprecated process.binding (`http_parser`) in node

Version 1.4.0

Release Date: September , 2024

Upgrades have been made to some third-party components delivered with XPP RESTful Web Service, including NodeJS, which has been updated to version 20.16.0, as well as dependencies uuid, tar, mocha, commander.

Fixed Issues

Issue #	Description
CRQ-39152	404 error when attempting to print using XPP RESTful Web Services API call due to length of path

Version 1.3.0

Release Date: June, 2024

Upgrades have been made to some third-party components delivered with XPP RESTful Web Service, including NodeJS, which has been updated to version 20.13.1.

Fixed Issues

Issue #	Description
XPP-8903	vulnerabilities reported against xpprest: semver 5.7.1
XPP-8668	vulnerabilities reported against xpprest: semver 6.3.0
XPP-8906	vulnerabilities reported against xpprest: semver 6.3.0
XPP-8904	vulnerabilities reported against xpprest: semver 7.3.5
XPP-8907	vulnerabilities reported against xpprest: semver 7.5.1
XPP-10087	vulnerabilities reported against xpprest: tar.6.1.15
XPP-8908	vulnerabilities reported against xpprest: yargs-parser 10.1.0

Version 1.2.0

Release Date: March, 2024

Upgrades have been made to some third-party components delivered with XPP RESTful Web Service, including NodeJS, which has been updated to version 20.10.0.

Fixed Issues

Issue #	Description
CRQ-36280	XPP RESTful Web Services silent install exits before finishing
XPP-6984	vulnerabilities reported against xpprest: json-schema:0.2.3
XPP-6985	vulnerabilities reported against xpprest: nanoid:3.1.25

Issue #	Description
XPP-6986	vulnerabillities reported against xpprest: node-windows:1.0.0-beta.5
XPP-7201	vulnerabilities reported against xpprest: async:3.2.1
XPP-7202	vulnerabilities reported against xpprest: minimist:1.2.5
XPP-7203	vulnerabilities reported against xpprest: moment:2.29.1
XPP-7484	vulnerabilities reported against xpprest: node.js
XPP-7672	vulnerabilities reported against xpprest: minimatch:3.0.4
XPP-7769	vulnerabilities reported against xpprest: zlib-pool.js
XPP-7817	vulnerability reported against xpprest: qs:6.10.1
XPP-7827	vulnerabilities reported against xpprest: extend-node.js
XPP-7828	vulnerabilities reported against xpprest: gatsby-node.js
XPP-7879	vulnerabilities reported against xpprest: jsonwebtoken:8.5.1
XPP-7966	vulnerability reported against xpprest: cookiejar:2.1.1
XPP-8220	vulnerability reported against xpprest: xml2js:0.4.23
XPP-8272	vulnerability reported against xpprest: bootstrap.min.js

Version 1.1.0

Release Date: November, 2021

Updated node.js to version 14.7.5 for enhanced security

Added Cross-origin Resource Sharing (CORS) support for all servers with an option to disable sharing from all external servers or limit access to a specified list of servers.

Fixed Issues

Issue #	Description	Notes
CRQ-24729	Correct XPP RESTful API User command documentation	The online XPP RESTful API documentation for the User command now reflects the correct command location: **`xz/xpprest/user`** and an example (`command=testws.pl`) was added to the parameter description.
XPP-5981	Fix documentation links in XPP RESTful API doc	Broken links to some XPP manuals were repaired in the online XPP RESTful API documentation.

1 Release Notes

2

Installing XPP RESTful Web Services

Installing or Upgrading XPP RESTful Web Services on Windows

This section describes prerequisites to installing or upgrading XPP RESTful Web Services on Windows, and the installation procedure itself.

Prerequisites for Installing

Before you install XPP RESTful Web Services, verify that the following prerequisites are met:

- XPP 9.4 or later is installed.
- The `default` XPP graphics library **`%XYV_EXECS%\distr\graphics`** is available.
- The *%XYV_EXECS%* environment variable is set.

Tip: To determine if the environment variable is defined, at the command prompt, type `set XYV_EXECS` and press **Enter**. The system should return the variable with its path. If not, the variable has not been set.

Before Upgrading

If XPP RESTful Web Services have already been installed previously, please make sure its service is not running before commencing with the upgrade.

Open a command prompt (cmd.exe) with “Run as Administrator” and execute the following command:

```
sc stop xpprest.exe
```

Alternately, you can perform the same action with a GUI.

1. Open **Control Panel > Administrative Tools > Services** .
2. Right-click **XPP Rest** and select **Stop** from the pop-up contextual menu.

Installing or Upgrading XPP RESTful Web Services

To install XPP RESTful Web Services:

1. Log in to the server as a user with administrative privileges.
2. To mount the installation ISO file, rght-click the file and select **Mount**.

Windows mounts the ISO file to the next available drive letter. Alternatively, you can use a third-party tool to mount the ISO as a disk.

3. Navigate to the location of the installation program:

   drive:\WINDOWS\VM

   (where *drive* is the drive where the installation ISO file was mounted).

4. Click the INSTALL.EXE executable or type INSTALL.
5. In the **Introduction** panel, click **Next**.
6. In the **License Agreement** panel, click I accept the terms of the License Agreement, and then click **Next**.
7. For a new install, a dialog will appear prompting you for the port number. The default port is 2995; optionally, enter a new number to change the port. Click **OK**.
8. In the **Pre-Installation Summary** panel, review your settings. If the settings are correct, click **Install**. The installation program displays a **Post-Installation Status** panel.
9. Following installation, the following XPP files and directories are created on your system:
   - XPP RESTful Web Services programs are installed to: **%XYV_EXECS%\xpprest**
   - XPP RESTful Web Services test scripts are installed to: **%XYV_EXECS%\xpprest\t**
   - Installer details are recorded in the log located at: **%XYV_EXECS%\xpprest\logs**
10. If this XPP server will service Windows clients, the document paths have been configured to use UNC path names. Because XPP RESTful is a service, it needs network read/write access, which the default Local System is not sufficient to allow.

    Do the following if the XPP server has been configured with Windows clients:

    a. In Windows, open **Control Panel > Administrative Tools > Services** .

    b. Right-click **XPP Rest** and select Properties.

    c. On the Properties dialog Log On tab, select **This account** and enter an account name with sufficient network file access privileges and its password.

    d. Restart the XPP RESTful service.
11. Validate the installation.

Console or Silent Installation

XPP RESTful can be installed without a GUI in either console or silent mode.

In console or silent mode, the port number defaults to 2995. If a different port is required, set it at the command line using the -DWS_PORT=xxxx option (where xxxx is the port number).

Installing in Console Mode

To install XPP RESTful Web Services in console mode, (which displays installation results), execute the installation program with the `-i console` option.

Example

The following command installs XPP RESTful Web Services in console mode using port `2996`:

```
INSTALL -i console -DWS_PORT=2996
```

Installing in Silent Mode

To install XPP RESTful Web Services in silent mode, execute the installation program using the `-i silent` option.

Example

The following command installs XPP RESTful Web Services in silent mode using port `2996`:

```
INSTALL -i silent -DWS_PORT=2996
```

Installing or Upgrading XPP RESTful Web Services on Linux

This section describes prerequisites to installing XPP RESTful Web Services on Red Hat Enterprise Linux, and the installation procedure itself. It also includes procedures for upgrading or uninstalling Web Services.

Prerequisites for Installing

Before installing XPP RESTful Web Services, verify that the following prerequisites are met:

- XPP 9.4 or later is installed.

  Important: Beginning with version 1.3.0, XPP RESTful Web Services is no longer supported on RHEL 7.

- The `default` XPP graphics library **`$XYV_EXECS/distr/graphics`** is available.
- The *$XYV_EXECS* environment variable is set.
  To do this, at a prompt, enter:
  `source /etc/xyvision/xyv.cshrc` or
  `source /etc/xyvision/xyv.profile`

Before Upgrading

If XPP RESTful Web Services have already been installed previously, please make sure its service is not running before commencing with the upgrade.

Execute the following shell command:

```
sudo systemctl stop xpprest
```

Installing or Upgrading XPP RESTful Web Services

To install XPP RESTful Web Services:

1. Log on as root.
2. Mount the installation ISO file.
   a. Type `sudo mkdir /mount-point`
   a. Type `sudo mount -t iso9660 -o loop cdrom.iso /mount-point` (where `cdrom.iso` is the name of the ISO file)
3. Enter the full path to the installation program:
   `sudo /mount-point/LINUX/VM/INSTALL.BIN`
1. In the **Introduction** panel, click **Next**.
2. In the **License Agreement** panel, click `I accept the terms of the License Agreement`, and then click **Next**.
3. For a new install, a dialog will appear prompting you for the port number. The default port is `2995`; optionally, enter a new number to change the port. Click **OK**.
4. In **Pre-Installation Summary** panel, review your settings. If the settings are correct, click **Install**. The installation program displays a post-installation status panel.
5. Click **Done**.
6. Following installation, the following XPP files and directories are created on your system:
   - XPP RESTful Web Services programs installed to **`$XYV_EXECS/xpprest`**
   - The XPP RESTful Web Services test scripts are installed to **`$XYV_EXECS/xpprest/t`**
   - Installer details are recorded in the log located at: **`$XYV_EXECS/xpprest/logs`**
7. Validate the installation.

Console or Silent Installation

XPP RESTful Web Services can be installed without a GUI in either console or silent mode.

In console or silent mode, the port number defaults to `2995`. If a different port is required, set it at the command line using the `-DWS_PORT=xxxx` option (where `xxxx` is the port number).

Installing in Console Mode

To install XPP RESTful Web Services in console mode, (which displays installation results), execute the installation program with the `-i console` option.

Example

The following command installs XPP RESTful Web Services in console mode using port `2996`:

```
INSTALL.BIN -i console -DWS_PORT=2996
```

Installing in Silent Mode

To install XPP RESTful Web Services in silent mode, execute the installation program using the `-i silent` option.

Example

The following command installs XPP RESTful Web Services in silent mode using port `2997`:

```
INSTALL.BIN -i silent -DWS_PORT=2997
```

Restarting the Service After Upgrading

If you stopped your system service for XPP REST prior to upgrading, you will want to start the service again.

Starting the Windows Service

Open a command prompt (**`cmd.exe`**) with “Run as Administrator” and execute the following command:

```
sc start xpprest.exe
```

Alternately, you can perform the same action with a GUI.

1. Open **Control Panel > Administrative Tools > Services** .
2. Right-click **XPP Rest** and select **Start** from the pop-up contextual menu.

Starting the Linux Service

Execute the following shell command:

```
sudo systemctl start xpprest
```

Uninstalling XPP RESTful Web Services

To uninstall XPP RESTful Web Services:

1. Log in as root on Linux or as Administrator on Windows.
2. From a command prompt, execute the uninstall command by entering the full path to the command:
   - Windows:

     `%XYV_EXECS%\xpprest\uninstall [-s] [n] [-h]`
   - Linux:

     `$XYV_EXECS/xpprest/uninstall [-s] [-n] [-h]`

     where:
     - `-s` suppresses the confirmation prompt and begins uninstalling the XPP RESTful Web Services software
     - `-n` displays what the uninstall program would do, but does not uninstall the software
     - `-h` displays the usage of the uninstall command

   Note: The switches for the uninstall command are optional.

3. If you executed the uninstall command without the `-s` switch, XPP prompts to confirm you want to continue uninstalling the software. Type `Y` to continue. The uninstall program stops the XPP RESTful Web Service and removes its folder and all of its files.

   Note: On Windows, you may see a message stating that the folder is not empty. If this occurs, run the uninstall program again.

Validating the Installation

The following steps let you validate the XPP RESTful Web Services installation or upgrade on Windows or Linux.

To test the XPP RESTful Web Services installation:

In a Web browser, type:

`http://host:port/index.html`

You should see an XPP RESTful Web Services Welcome page, which includes information about versions of XPP RESTful Web Services and XPP, details on the dependent NodeJS libraries and their versions, as well as a link to the API documentation.

The API documentation is designed to be interactive, providing detailed descriptions of each REST endpoint, including their expected parameters and the responses you can anticipate. Built on top of the popular Swagger framework, our documentation makes it

easy to explore and interact with the API.

Verifying the Active XPP RESTful Web Services Version

To verify the active XPP RESTful Web Services version, run the following from the command line:

- Windows:
  ```
  %XYV_EXECS%\xpprest\node\node.exe %XYV_EXECS%\xpprest\server.js --version
  ```
- Linux:
  ```
  $XYV_EXECS/xpprest/node/bin/node $XYV_EXECS/xpprest/server.js --version
  ```

Identifying the Node.js Version

To verify the version of Node.js in XPP RESTful Web Services, run the following from the command line:

- Windows:
  ```
  %XYV_EXECS%\xpprest\node\node.exe --version
  ```
- Linux:
  ```
  $XYV_EXECS/xpprest/node/bin/node --version
  ```

XPP RESTful Web Services Test Suite

To test the XPP RESTful Web Services installation, a JavaScript test environment is provided. You can invoke the test command at a command prompt from the xpprest directory.

JavaScript Testing

To test that the JavaScript scripts are able to communicate with the XPP RESTful Web Services software:

1. From the command prompt (Windows) or an Xterm window (Linux), change to the **`%XYV_EXECS%\xpprest`** folder on Windows or the **`$XYV_EXECS/xpprest`** directory on Linux.
2. Enter the JavaScript test command: test-rest

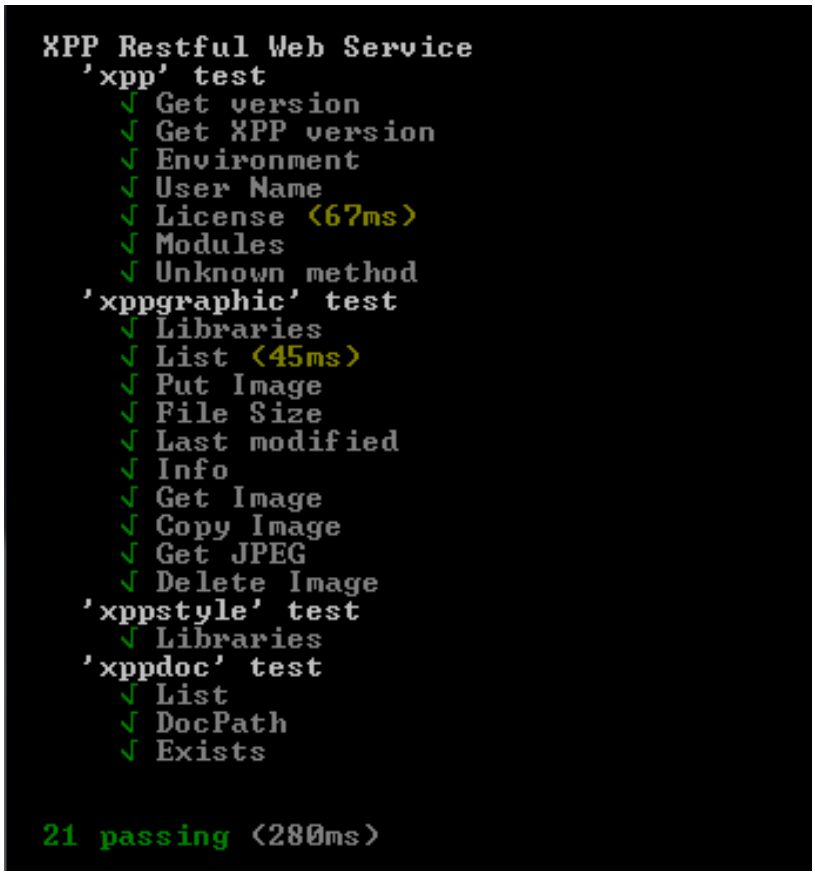
Note:

If the default protocol (http) and/or port (`2995`) are not used, add the `--host protocol://host:port` as an argument to the test-rest command, such as:

```
test-rest --host http://localhost:8443
```

If your XPP REST server uses HTTPS and/or requires a client certificate for authentication, see the section on "Testing with TLS" below.

The following are sample JavaScript test results:



Validating the Online Documentation

To verify that the online documentation has been successfully loaded:

1. In a web browse, type **`http://xppwebserver:####/index.html`** where `xppwebserver` is the name of the web server and `####` is the port number.
2. Press **Enter**. The XPP RESTful Web Services home page appears.
3. Click the **Documentation** link. The online documentation should appear.

Note:

The XPP RESTful Web Services online documentation refers to the "XPP Command Line Utilities Manual," which is not delivered with XPP RESTful Web Services. To link the references to the online documentation to that manual, copy the **cmdutils.pdf** file from XPP documentation downloadable at docs.rws.com to the following directory:

- Windows: `%XYV_EXECS%\xpprest\xpp\documentation`
- Linux: `$XYV_EXECS/xpprest/xpp/documentation`

Services and Logging

The XPP RESTful server is installed as a service on Windows and Linux. As a background server process, it supports implied functionality, including:

- Automatically starting when the system is booted
- Logging for the service process
- Ability to manually stop and start the service
- Error logging

XPP RESTful Server as a Service

Running the XPP RESTful server as a service is controlled differently on Windows and Linux.

Windows

On Windows, the XPP RESTful service is accessible through **Control Panel > Administrator Tools > Services** so that it can be managed like any other Windows service. Its display name is **XPP Rest** and its actual service name is **`xpprest.exe`**.

Start and stop messages from the XPP RESTful service are available through the Windows **Event Viewer** under the **Application** section.

Status and error messages are written to log files named **`xpprest.out.log`** and **`xpprest.err.log`**, respectively, which are located in the **`%XYV_EXECS%\xpprest\daemon`** folder.

Linux

On Linux, `xpprest.service` is installed and registered with the systemd environment and is available through the `systemctl` utility.

To validate the service is running, run:

```
systemctl status xpprest
```

To control the starting and stopping of the service, for any reason, run:

```
systemctl [start | stop | restart] xpprest
```

To view log messages specific to the service, run:

```
journalctl -u xpprest
```

XPP RESTful Socket Timeout

The XPP RESTful server has a timeout option to set the server socket timeout. If a process takes longer than the timeout, the connection will close and the user will receive a timeout error code. The default timeout for any URL function is six hours.

A single XPP web service API call should not take more than six hours. However, if the timeout value needs to be changed, a `--timeout value` option (where `value` is the number of seconds) can be added to the configuration file in a manner similar to changing the port number. See "Post-installation Configuration" for information on how to add or change option arguments when starting the XPP RESTful service.

Note: When calling the XPP RESTful API through a client, the client-side connection may have its own timeout. This must be changed at the client connection level.

2 Installing XPP RESTful Web Services

3

Configuration Options

This chapter describes the following optional configurations:

- Changing the port number
- Using the Secure Socket Layer (SSL) protocol
- Configuring user access
- Cross-origin Resource Sharing (CORS)
- Configuring client certificate authentication

Changing the Port Number (Optional)

If you need to change the port number of the XPP RESTful server, a service configuration file can be modified post-installation.

An environment variable, `XYV_REST_IARGS`, is used to pass configuration options to the run-time server. The `-port xxxx` option is used to specify the port. The process for changing the port and restarting the service is dependent on the operating system.

Changing the Port on Windows

To change the port number on a Windows system:

1. Stop the XPP RESTful service using the **Services Control Panel**.
2. Edit the **`xpprest/daemon/xpprest.xml`** file:
   a. Find the line:
   ```
   <env name="XYV_REST_IARGS" value="--port 2995"/>
   ```
   b. Change the port to the appropriate number, as in:
   ```
   env name="XYV_REST_IARGS" value="--port 3007"/>
   ```
3. Start the service using the **Services Control Panel**.

Changing the Port on Linux

To change the port number on a Linux system:

1. Stop the XPP RESTful service using `systemctl stop xpprest`
2. Edit the **`/etc/systemd/system/xpprest.service`** file:
   a. Look for the line containing:
   ```
   --env XYV_REST_IARGS=\"--port 2995\"
   ```
   b. Change the port to the desired number; as in:
   ```
   --env XYV_REST_IARGS=\"--port 3007\" ...
   ```
3. Any time a **`systemd`** service file, such as **`xpprest.service`**, is updated, you must perform:
   ```
   systemctl daemon-reload
   ```

4. Start the service using:
   ```
   systemctl start xpprest
   ```

Using the Secure Socket Layer (SSL) Protocol (Optional)

The XPP RESTful Web Services server does not require SSL. Since it is a low-level API, it is designed to sit on top of an XPP server and it is assumed that any application that requires XPP RESTful Web Services will provide its own functionality within its web server. However, due to security concerns, you may want all server socket traffic to be encrypted with SSL. To meet this requirement, the server can be configured to run with SSL.

SSL setup consists of providing a certificate and private key files in the "pem" format. When installing the service, a `--cert filename` option must be configured on the XPP RESTful Web Services server. This option looks for the two 'pem' files in the **`xpprest/ssl/`** directory as **`name.pem`** (certificate) and **`name-key.pem`** (private key).

When SSL is configured, XPP REST Web Services automatically enables additional security features:

- HTTP Strict Transport Security (HSTS) headers are automatically added to all responses, instructing browsers to only connect via HTTPS and helping prevent man-in-the-middle attacks.
- Strong cipher suites are enforced, with weak encryption protocols automatically disabled to meet modern security standards.

Because this process is outside the standard installation and delivery, the installion program does not handle SSL natively; instead, SSL setup is an optional post-installation process. The following sections describe this process for Windows and Linux systems.

Setting Up SSL on Windows

To configure XPP RESTful Web Services to use SSL on a Windows system:

1. Ensure the two 'pem' files are placed in the **`%XYV_EXECS%\xpprest\ssl`** folder as **`name.pem`** and **`name-key.pem`** (where `name` is the base name of the file(s) that contain the security keys).
2. Stop the XPPrest service using the **Services Control Panel**.
3. Edit the **`xpprest\daemon\xpprest.xml`** file:
   a. Find the line:
      ```
      <env name="XYV_REST_IARGS" value="--port 2995"/>
      ```
   b. Add to the value attribute `--cert name` (where `name` is the base name of the certificate 'pem' files), as in:
      ```
      <env name="XYV_REST_IARGS" value="--port 2995 --cert name"/>
      ```
4. Start the service using the **Services Control Panel**.

Setting Up SSL on Linux

To configure XPP RESTful Web Services to use SSL on a Linux system:

1. Ensure the two 'pem' files are placed in the **`$XYV_EXECS/xpprest/ssl`** folder as **`name.pem`** and **`name- key.pem`** (where `name` is the base name of the file(s) that contain the security keys).
2. Stop the XPP RESTful service using `systemctl stop xpprest`
3. Edit the **`/etc/systemd/xpprest.service`** file:
   a. Look for the line containing:
      ```
      ... --env XYV_REST_IARGS=\"--port 2995\"
      ```
   b. Add to the enviroment variable definition `--cert name` (where `name` is the base name of the certificate 'pem' files), as in:
      ```
      ... --env XYV_REST_IARGS=\"--port 2995 --cert name\" ...
      ```
4. Reload the `systemd` service:
   ```
   systemctl daemon-reload
   ```
5. Start the service using:
   ```
   systemctl start xpprest
   ```

Testing with TLS

For detailed instructions on testing the XPP RESTful service on the `https` protocol, refer to "Validating the Install" section.

Because this protocol is not the default, you must add the argument, **`protocol://host:port`** to the JavaScript test command, as in the following examples.

Testing SSL on Linux

```
cd $XYV_EXECS/xpprest
./test-rest --host https://hostname:portnumber
```

With client certificate authentication enabled:

```
cd $XYV_EXECS/xpprest
./test-rest \
  --host https://hostname:portnumber \
  --client-key path/to/client-key.pem \
  --client-cert path/to/client-cert.pem
```

Testing SSL on Windows

```
cd %XYV_EXECS%\xpprest
test-rest.cmd --host https://hostname:portnumber
```

With client certificate authentication enabled:

```
cd %XYV_EXECS%\xpprest
test-rest.cmd ^
  --host https://hostname:portnumber ^
  --client-key path\to\client-key.pem ^
  --client-cert path\to\client-cert.pem
```

Verifying Security Features When testing with SSL enabled, you can verify that security features are working correctly:

- Check that HSTS headers are present by examining the response headers for `Strict-Transport-Security`.
- Verify that only strong cipher suites are in use by using SSL testing tools such as OpenSSL or online SSL analyzers.

```
openssl s_client -connect yourdomain.com:443 -cipher ECDHE-RSA-AES256-GCM-SHA384
```

Configuring User Access (Optional)

The XPP RESTful Web Services server is installed on the same system as the XPP server. Since the API is an application programming interface, it is assumed that any client application using the API will provide its own security on the front end of its application if required. For example, a web application would require its own login procedure prior to connecting to the XPP RESTful server.

By default, the XPP RESTful server requires registering a user prior to using the API. In this way, a log file captures the active users, and individual API functions are not available unless the user has registered with the XPP RESTful service. While not secure (since it doesn't require name/password validation), it at least allows you to track who is using the service and control random access to the individual API calls.

Security Recommendation:

For production environments, it is strongly recommended to combine user registration with SSL configuration to ensure that registration tokens are transmitted securely and cannot be intercepted.

Access to the API is available through a registration endpoint, **`http://host:port/xpp/register`** and the caller must provide a JSON object in the HTTP body of the form `{ name:`

`"User Name", email: "name@xxxx.com" }.`

Failure to provide the `name` and `email` properties will return a "BadRequest" error. The HTTP response will be a JSON object in the HTTP body containing a token as a long unique string of the form `{ token: "JFLajfakFA:LLAK:LD......." }`. After retrieving the token, the user must set the HTTP "Authorization" header on every subsequent API call as:

```
Authorization: Bearer JFLajfakFA:LLAK:LD........
```

Failure to set the header to a proper value will result in a "401 Unauthorized" HTTP status return code with a JSON object returned as:

```
{"code":"InvalidCredentials","message":"No authorization token was found"}.
```

The **`/index.html`**, **`/xpp/documentation/*`**, **`/xpp/ping`**, and **`/xpp/register`** endpoints do not need a registration token for access.

It is possible to completely disable the registration process. Doing so will allow access to all API calls without requiring an Authorization header. To do so, refer to the "Changing the Port Number (Optional)" section for sample instructions for adding an option and add a `--noregister` option. Restarting the service will disable registration and the **`/xpp/register`** setup will not have to be invoked.

Cross-origin Resource Sharing (CORS)

Cross-origin Resource Sharing (CORS) is an HTTP-header-based mechanism that allows a server to indicate any origins (such as a domain, scheme, or port) other than its own from which a browser should permit loading of resources.

CORS relies on a mechanism by which a browser makes a "preflight" request to the server hosting the cross-origin resource to check that the server will permit the actual request. In the preflight request, the browser sends headers specifying the HTTP method and headers that will be used in the actual request.

XPP RESTful Web Services is automatically configured to allow CORS cross-site references. You can prevent any web page from accessing the XPP RESTful API or specify a list of known servers to be permitted access to the API. To do so, add an `-access value` option to the XPP RESTful service configuration file and set `value` as follows:

- To disable all CORS access, set `value` to `NONE`
- To specify a list of servers to be granted access, set `value` to the name of a file in the **`xpprest`** folder containing the server names, one per line.

Refer to the "Changing the Port Number (Optional)" section - for sample instructions for adding an option.

Enabling Client Certificate Authentication (Optional)

XPP RESTful services can be configured so that connections are required to provide a client certificate for authentication. The connection's certificate will be validated against a certificate authority (CA) X.509 certificate file that is provided to XPP REST when it starts up. Connections with a valid certificate, signed by this CA, are allowed to use XPP REST. Clients lacking this will fail to connect.

SSL Requirement

In order to effectuate this security feature, XPP RESTful services must be configured with SSL for its server. Naturally, client connections will have to use HTTPS (in addition to sending their client certificate) in order to connect. Please consult the preceding chapter on setting up SSL.

Server Configuration

The server configuration for client certificate authentication is similar to SSL. Your certificate authority (CA) X.509 file (in `pem` format) must be in same **`ssl`** subdirectory where XPP REST is installed. Once this file has been copied there, you need to specify the base of this file (minus the `pem` extension) to XPP REST using the `--cert-auth` option. For example, if your CA filename is **`excalibur.pem`**, this option would be expressed like this `--cert-auth excalibur`

Windows Configuration Instructions

1. Ensure your certificate authority X.509 certificate file (in `pem` format) is in the **`%XYV_EXECS%\xpprest\ssl`** subfolder.
2. Stop the **XPP Rest** service using the Microsoft Management Console (**MMC**) for Services.
3. Open the **`%XYV_EXECS%\xpprest\daemon\xpprest.xml`** file in your preferred text editor.
4. Find the line containing `XYV_REST_IARGS`, like this:

```
<env name="XYV_REST_IARGS" value="--port 2995 --cert name"/>
```

5. Add `--cert-auth` with the base name of your CA's `pem` file to the `value` attribute:

```
<env name="XYV_REST_IARGS" value="--port 2995 --cert crt-base --cert-
auth ca-base"/>
```

6. Start the **XPP Rest** service using **MMC**.

Linux Configuration Instructions

1. Ensure your certificate authority X.509 certificate file (in `pem` format) is in the **`$XYV_EXECS/xpprest/ssl`** subfolder.
2. Stop the **XPP Rest** service using `systemctl stop xpprest`.
3. Open the **`/etc/systemd/xpprest.service`** file in your preferred text editor.
4. Look for the line containing:

```
--env XYV_REST_IARGS="--port 2995 --cert crt-base"
```

5. Add `--cert-auth` with the base name of your CA's `pem` file inside the double quotes:

```
--env XYV_REST_IARGS="--port 2995 --cert crt-base --cert-auth ca-base"
```

6. Reload the `systemd` service with `systemctl daemon-reload`.
7. Start the service using `systemctl start xpprest`.

Testing Client Certificate Authentication

After you have configured XPP REST's certificate authority, clients will need a valid client certificate signed by this CA in order to connect. The client certificate must be in `pem` format with the key and certificate concatenated into one file. For the example below, this certificate file is called **`my-client-certificate.pem`**

Test Command

```
curl --cert my-client-certificate.pem https://localhost:2995/xpp/ping
```

You should see `{"message":"ping"}` as a result.