Introduction to Crestron Fusion® Software API

Enterprise Management Platform

Getting Started

Crestron Electronics, Inc.

This document was written by the Technical Publications department at Crestron.

Contents

Introduction 1

Resource Data Web Service 1
External Web Service 1

Development Resources 2

Help Files 2
Crestron Fusion API CHM Help File Location 2
Endpoint Operation Reference 2
Sample Code 2
.NET API Test Harness 2
Java® Software Test Harness 3
External Web Service 3
.NET Code 3

Configuring the External Web Service 3

Configuring System-Wide Settings 3
Setting a Room Calendar to Use an External Provider 4
Gateway Security Options 4

Configuring the Resource Data Web Service 5

Enable or Disable API 5
Security Options 5
Unencrypted Security Token 6
Encrypted Security Token 6
No Security 6
Testing the Resource Data API 6

Setting Up a Development Environment 7

Tools for Creating a Resource Data Web Service Client Application 7
Microsoft .NET 7
Java and JavaScript 7
Other Platforms 7
Tools for Developing Integration with the External Web Service 7
Developing a Client Application for the Resource Data Web Service 8
Creating an API Client Using Microsoft .NET 8
Creating a Java Client to Communicate with XML 8
Walk-Through Using the .NET Data API Test Harness Sample Code 8
Developing Integration with the XWS 9
Interfacing with an External Scheduling Provider 9
Walk-Through Using the Scheduling Provider Examples 9

Introduction to Crestron Fusion Software API: Enterprise Management Platform

Introduction

This document serves as an introduction to the Crestron Fusion® Application Programming Interface (API). It includes references to online help, sample code, and appropriate utilities. This document is intended for programmers.

Resource Data Web Service

The Resource Data Web Service provides Representational State Transfer (REST) resources and Create, Read, Update, Delete (CRUD) operations for the Crestron Fusion objects exposed by the API. Data can be exchanged in either Extensible Markup Language (XML) or JavaScript Object Notation (JSON) formats. The platform is interoperable with any client platform capable of Hypertext Transfer Protocol (HTTP) communication over Transmission Control Protocol (TCP). The REST Uniform Resource Identifier (URI) is documented in the Crestron.Fusion.API.Resources namespace.

The Resource Data Web Service features are as follows:

- GET MANY operations are available for each resource. Resources include Actions, Appointments, Assets, Attributes, Availability, Nodes, Rooms, and Signalvalues. These operations provide filtered, paged, and lightweight versions of the resource objects for browse lists.
- Full CRUD maintenance on the room, asset, action, and appointment resource is provided. Attribute and node resources are read-only.
- Send operations are available for the action and attribute resources to allow control of Crestron® devices.

External Web Service

The External Web Service (XWS) is an interface exposed by Crestron Fusion that allows External Calendaring Applications (XCA) to integrate with Crestron Fusion room scheduling.

The XWS features are as follows:

- Full synchronization of appointment information between Crestron Fusion and the XCA is provided
- Customizable sample web service code is provided to assist in integrating an external scheduling application via REST over HTTP

Development Resources

A variety of development resources are included with each installation. They can be found both at the URL of the Crestron Fusion installation and in Internet Information Server (IIS) when the user is logged into the Crestron Fusion server.

Help Files

Help files are provided with every Crestron Fusion installation. A website describing each resource and operation is available at http://<*CrestronFusionServer*>/fusion/apiservice/help.

Crestron Fusion API CHM Help File Location

A compiled HTML help (*.chm) file is available at http://<*CrestronFusionServer*>/fusion/apiservice/help.

When logged into the Crestron Fusion server, locate the HTML help at <*install drive*>\Program Files\Crestron\Fusion\FSM\Instances\Fusion\APIHelp\Crestron_Fusion_API.chm.

NOTE: Due to possible browser and local network restrictions, it is recommended to save the HTML help file to a local drive or desktop.

Endpoint Operation Reference

Details of the data exchanged with the API server are available at http://<*CrestronFusionServer*>/fusion/apiservice/help.

NOTE: Substitute <*resource*> with the resource name (i.e., rooms or assets).

Sample Code

Sample code is provided for a variety of platforms, classes, and helper utilities.

.NET API Test Harness

A sample application demonstrating how to integrate with the Crestron Fusion API is available at <*install drive*>\Program Files\Crestron\Fusion\FSM\API_Samples\API_NET_Samples.zip.

NOTE: For more information on this application, refer to “Creating an API Client Using Microsoft .NET” on page 8.

Java® Software Test Harness

A sample application demonstrating an advanced marshalling concept integrated with the Crestron Fusion API is available at <*install drive*>\Program Files\Crestron\Fusion\FSM\API_Samples\API_JAVA_Samples.zip.

NOTE: For more information on this application, refer to "Creating a Java Client to Communicate with XML" on page 8.

External Web Service

A sample application for the XWS can be found at <*install drive*>\Program Files\Crestron\Fusion\FSM\API_Samples\ ExternalWebService_Samples.zip.

NOTE: For more information on this application, refer to "Configuring the External Web Service" on page 3.

.NET Code

Examples and code for shared entity classes and API helper utilities are provided with each installation.

A binary containing Crestron Fusion API resource models (also known as entities) is available for reuse at <*install drive*>\Program Files\Crestron\Fusion\FSM\Instances\Fusion\ApiService\bin\ ApiHelper.dll.

The source code for the APIHelper.DLL can be found at <*install drive*>\Program Files\Crestron\Fusion\FSM\API_Samples\API_NET_Samples.zip.

Reusable code to perform XML Serialization, logging, and encryption is available for client applications to copy or reference at <*install drive*>\Program Files\ Crestron\Crestron Fusion\FSM\ API_Samples\API_NET_Samples.zip.

Configuring the External Web Service

The following procedures are provided to set up the XWS to communicate with external groupware and scheduling providers.

Configuring System-Wide Settings

To set up the XWS gateway and login credentials in the Crestron Fusion Configuration Web Client, follow the steps below.

1. Open the Crestron Fusion Configuration Web Client.
2. Select the **Configuration** tab at the bottom left of the screen.
3. Select the **Scheduling** folder in the tree view.
4. Select the **External Settings** tab.
5. In the **External Web Service Default URL** field, enter the base Uniform Resource Locator (URL) for the desired XWS. The format of the base URL is http://<*servername*>/<*virtualdirectory*>.

6. Enter a username and password in the corresponding text boxes. The login credentials are presented to the third-party scheduling application by the gateway service for each room's calendar unless they are overridden for the room as described in step 5 of "Setting a Room Calendar to Use an External Provider" below.

Setting a Room Calendar to Use an External Provider

To link a room calendar to an external groupware provider in the Crestron Fusion Setup Web Client, follow the steps below.

1. Open the Crestron Fusion Setup Web Client.
2. Locate the room designated to use the external provider and open it for editing.
3. On the **Scheduling Details** tab, select **External** from the **Server Access** drop-down menu.
4. Enter the proper identifier for the calendar to be associated with the room in the **Calendar ID** field. The format of this identifier is dependent on the third-party scheduling application. It is saved as a simple text string in the Crestron Fusion database.
5. If it is necessary to override the gateway service URL set in step 5 of "Configuring System-Wide Settings" on page 3, enter the desired alternate URL in the **URL** field.
6. If it is necessary to override the login credentials for the third-party scheduling application set in step 6 of "Configuring System-Wide Settings" on page 3, enter the desired username and password in the appropriate text boxes.

Gateway Security Options

The XWS uses the same mechanism to protect traffic between itself and the Crestron Fusion Scheduling and Groupware services as the Resource Data Web Service uses for itself and its clients. For more information, refer to "Security Options" on page 5.

Security for the XWS can be provided by the following mechanisms used individually or in combination:

- The API client passcode that is established in the Configuration Web Client is used to prevent casual HTTP requests from being handled by the XWS.
- HTTPS can be implemented between Crestron Fusion services and the XWS.
- Additional encryption of the URL components can be enabled between Crestron Fusion services and the XWS by selecting the appropriate options in the Crestron Fusion Configuration Web Client.

For initial development and testing, it is recommended to turn off all security mechanisms; however, since the credentials for the third-party calendaring application must be transferred to the gateway, the password must always be encrypted regardless of the configuration settings.

The security settings for the gateway must be set in its app.config file in the **appSettings** section with the SecurityLevel key. The value of that key corresponds to the configuration settings in the table below.

SecurityLevel Key Settings

ENABLE API SECURITY	ENCRYPT SECURITY TOKENS	SECURITYLEVEL VALUE
Unchecked	Unchecked	NoSecurity
Checked	Unchecked	ClearToken
Checked	Checked	EncryptedToken
Unchecked	Checked	(not defined)

Configuring the Resource Data Web Service

Use the procedures in this section to configure the Resource Data Web Service.

Enable or Disable API

The API is disabled by default. To enable the API, complete the steps below:

1. Navigate to the Crestron Fusion Configuration Web Client.
2. Select the **Configuration** tab in the lower left corner.
3. Select the **API folder** in the left-hand navigation tree.
4. Check **Enable API Service** and click **Save**.
5. Perform an IISRESET after enabling the API Service.

To disable the Crestron Fusion API, complete the steps below:

1. Navigate to the Crestron Fusion Configuration Web Client.
2. Select the **Configuration** tab in the lower left corner.
3. Select the **API folder** in the left-hand navigation tree.
4. Uncheck **Enable API Service** and click **Save**.

NOTE: The new configuration is reloaded after an IIS restart or recycle is performed.

Security Options

The Crestron Fusion API service relies on the client application for authentication. Instead of authenticating users, the API Service authenticates the client application using a simple shared passcode. This passcode is set in the API section of the Configuration Web Client. Three security options are available: unencrypted security token, encrypted security token, and no security.

Unencrypted Security Token

This level of security is set when the **Enable API Security** check box is checked and the **Encrypt Security Tokens** check box is unchecked. This configuration must never be used without implementing Secure Socket Layer (SSL).

Under this configuration, the client applications request a security token from the API Service that is sent on all requests. The security token request must contain a set of comma-separated Crestron Fusion role names to access object-level security. The Crestron.Fusion.API.Security namespace contains details on the URI. The token is not encrypted and is susceptible to replay attacks without SSL.

Encrypted Security Token

Encrypted security tokens are necessary when the Crestron Fusion API Service is not using SSL to prevent replay attacks from malicious requests. This level of security is set when the **Enable API Security** check box is checked and the **Encrypt Security Tokens** check box is checked.

After receiving a security token from the API Service, the client application must perform the following:

1. Decrypt the security token using the shared passcode and save the user ID.
2. Encrypt the user's ID and the current UTC date and time in RFC3999 format (before every subsequent request to the server). The algorithm is MD5.
3. Set the encrypted string to the ?auth= query string variable in the request URI.

Upon receiving the request, the server decrypts the security token and compares it with the time it was sent to the server to the server time adjusted for UTC. The number entered in the **Token Timeout** field determines the number of seconds the server time can differ before the request is rejected.

No Security

Although it is never advised in production, security can be turned off for development purposes by un-checking the **Enable API Security** check box. All client requests then impersonate the default administrator account.

Testing the Resource Data API

After enabling the API with no security, verify the functionality by opening a browser and navigating to http://<*server-name*>/fusion/apiservice/rooms. A list of rooms is displayed that are in Crestron Fusion. If there is no response from the Crestron Fusion API, contact the Crestron Fusion Support Group (FSG) at (855-754-5962) or e-mail fsg@crestron.com.

Setting Up a Development Environment

Tools for Creating a Resource Data Web Service Client Application

Multiple platforms may be used to create a client.

Microsoft .NET

For a client residing on a .NET web server, the following components are recommended:

- Microsoft® Visual Studio® 2010 software or later (other versions may be used after recreating sample *.sln files)
- Internet Information Server (IIS)

Java and JavaScript

For a client residing on a Java web server, the following components are recommended:

- NetBeans® IDE software (Community Edition may be used)
- GlassFish® Web Server

For JavaScript implementations, use the following to create a page-based client:

- jQuery and JSON/AJAX (Asynchronous JavaScript and XML) libraries, which utilize the loosely typed JSONObject class
- Any popular web page editor to assist in JavaScript® software development

For a forms-based Java client, the NetBeans IDE can be used without a web server.

Other Platforms

Any development platform with the following criteria can be used to create a client:

- The platform is able to send and receive HTTP messages over TCP.
- The platform is capable of handling XML or JSON without string manipulation.

NOTE: Although it is possible to create a client application that extracts data directly from strings, it is highly recommended that an XML serializer or JSON parser be used to create objects, as future updates to the API could impact your existing development.

Tools for Developing Integration with the External Web Service

Interfacing with the XWS requires binary reference Crestron Fusion internal DLLs, and a web service is recommended for interoperability. The following tools are recommended:

- Visual Studio (VS) 2010 or later that is used to reference Crestron Fusion and create the recommended web service.
- Platforms that are used to create a client for the recommended web service listed in “Tools for Creating a Resource Data Web Service Client Application” above.

Developing a Client Application for the Resource Data Web Service

An API client can be created using Microsoft.NET and a Java client can be created to communicate with XML.

Creating an API Client Using Microsoft .NET

Since the Crestron Fusion API Server is REST-based, the following .NET components are recommended for creating requests and interpreting server responses over HTTP:

- System.NET.WebRequest sends an HTTP resource operation request to the Crestron Fusion API server consisting of a URI, Verb, and Content. Set the ContentType property to JSON or XML.
- System.NET.WebResponse reads the HTTP responses to client requests. The responses from the Crestron Fusion API server typically consist of an HTTP status code and content wrapped in an API_Response object.
- System.XML.Serialization.XMLSerializer provides functionality to convert XML into a fully typed instantiated class that is based on entity models provided in the framework.
- System.Runtime.Serialization.Json.DataContractJsonSerializer provides functionality to convert JSON into a fully typed instantiated class that is based on entity models provided in the framework.

Usage of these classes is highlighted in the .NET test harness sample code in the JsonHelper and HTTPHelper classes.

Creating a Java Client to Communicate with XML

Java clients can send and receive HTTP messages with XML content (similar to the .NET API client) using the following libraries:

- Java Development Kit (version 1.7 minimum) from www.oracle.com/java provides the application foundation.
- HTTPClient (version 4.1.3) creates and reads messages over HTTP.
- Jaxen (version 1.1) provides XML parsing and XPath queries.

Usage of these libraries is demonstrated in the Java software test harness sample code in the InvokeAPIService methods of the servlet classes in the Source Packages folder.

Walk-Through Using the .NET Data API Test Harness Sample Code

The sample code for the Data API can be extracted from the following location on your Crestron Fusion server: <*install drive*>\Program Files\Crestron\Crestron Fusion\FSM\API_Samples\ API_NET_Samples.zip.

Follow the steps below to run the .NET test harness and inspect the code at runtime.

1. Navigate to the **Samples** folder on the Crestron Fusion Server and open the **ApiTestHarnessMicrosoft** subfolder.
2. Double-click **ApiTestHarnessMicrosoft.sln** to open the test harness in Visual Studio. Change the web.config app setting ServiceURL to the path of the ApiService installed in the development environment.

3. Open the **HTTPHelper** class in the **Helpers** folder of the **ApiTestHarnessMicrosoft** project. Refer to the application of the .NET classes in “Creating an API Client Using Microsoft .NET” on page 8. Set a breakpoint on the first line of each function.
4. Open the **XMLHelper** class in the **ApiHelper** project. Note the application of the XMLSerializer, which has functions to both serialize and deserialize XML.
5. Set a breakpoint on the first line of each function, from the **Model** objects contained in the **Entities** folder of the **ApiHelper** project.
6. Click **Run** and then click **Login**.

   NOTE: Crestron Fusion API configuration must be set to use security with no encryption as shown in “Security Options” on page 5.

7. Click the **Test All** tab and then click **Test All**. Advance through the code while inspecting the objects in the **Watch** window.

Developing Integration with the XWS

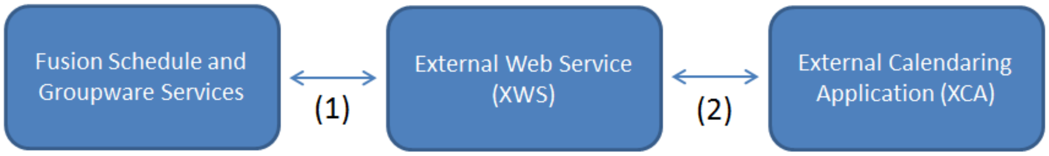
Interfacing with an External Scheduling Provider

The diagram below illustrates the data flow among the entities involved.

- Arrow 1 indicates an HTTP REST protocol, which has been defined by Crestron specifically for interaction with an XWS.
- Arrow 2 indicates an unknown protocol that exists between the XWS and External Calendaring Application (XCA).

The External Web Service is a translator that can be developed in most common languages to allow for interaction between the Crestron Fusion API Service and an External Calendaring Application.

Crestron Fusion and External Services Data Flow


Fusion Schedule and Groupware Services
(1)
External Web Service (XWS)
(2)
External Calendaring Application (XCA)

Walk-Through Using the Scheduling Provider Examples

To properly set up an XWS, which allows the XCA to become an appointment store for Crestron Fusion, follow the steps below.

1. Choose a host server where the XWS is to be installed. While it is possible to use one of the servers where Crestron Fusion runs, this is not recommended since, depending on load (number of rooms and types of Crestron equipment), this might not be an optimal choice. Also, the routing of calls to the XWS via HTTP is most efficient when carried to the XWS on a separate, remote server or on the XCA server itself.

   For the purposes of the sample solution, it is assumed that a .NET Web Service is to run under IIS. If this is not the preferred environment, go to step 5.

2. Edit the sample solution in Visual Studio 2010 or later. Find all references to the PhonyStore class in this project.

3. Code the replacement functions for the identified PhonyStore functions. Note that the ApiHelper project is included to provide a set of basic entity classes, which are used over the REST calls to and from Crestron Fusion, as well as utilities to convert, serialize, and encrypt strings sent between Crestron Fusion and the XWS.
4. Test the XWS using a simple HTTP test harness such as the Advanced Rest Client from the Google Chrome™ browser.
5. Deploy the XWS to the selected host server from step 1. Specify a virtual directory (e.g., MyXWS), and establish access via a user ID and password. Be sure the API Service is enabled for anonymous access.
6. Use the Crestron Fusion Configuration Web Client to access the **Scheduling** branch in the tree, and select the **External Web Service** tab.
7. Provide the base URL for the XWS in the **External Web Service** tab. For example, if the XWS is deployed on MyHost.xyz.com, under a virtual directory of MyXWS, the base URL is http://<MyHost.xyz.com>/<MyXWS>.
8. Provide default credentials for access to the XWS. If access to the XWS is not over a private Local Area Network (LAN), consider applying a security certificate and enter https://<MyHost.xyz.com>/<MyXWS>, as noted in the previous step.
9. Use the Crestron Fusion Setup Web Client to add a test room. Click **External** from the **Scheduling** tab drop-down menu. Enter the proper identifier for the calendar to be assigned to the room. The format of the identifier is entirely dependent on the XCA. Save the room. No further asset or symbol information is required until it has been verified that the XWS is working.
10. Use the appropriate client for the XCA to add an appointment to the calendar assigned to the room.
11. Use the Crestron Fusion Monitoring Web Client to view the room's schedule, and verify that the appointment appears at the correct date and time.
12. Add an appointment from the Crestron Fusion Web Client (if permitted). Then update and delete that appointment, again assuming that the XWS allows this.

This page is intentionally left blank.

Crestron Electronics, Inc.
15 Volvo Drive, Rockleigh, NJ 07647
Tel: 888.CRESTRON
Fax: 201.767.7576
www.crestron.com

Getting Started – DOC. 7706E
(2037275)
01.18
Specifications subject to
change without notice.