

# HTTPS Connector Setup

Download the PDF of this article.

## In this Article

- [Introduction](#)
- [Requirements](#)
- [Remote Script & Authentication](#)
- [Field Mapping](#)
- [Activity Log](#)
- [Create an Attachment from a Response](#)
- [POST Response Alias](#)
- [Video Tutorials](#)

## Related Articles

**July 2023 - Note:** A recent change to FormAssembly's file storage system may result in a 302 Redirect Error when using the HTTPS connector to retrieve file uploads. Now, when using the HTTPS connector to retrieve file uploads, a new redirect step takes place. To avoid a 302 Redirect Error, your scripts may need to be updated.

## Introduction

The HTTPS connector allows you to automatically forward submitted data from FormAssembly to any remote script using the standard [HTTP POST](#), [HTTP GET](#), or HTTP PUT methods.

When using the HTTPS connector, data is sent as application/x-www-form-urlencoded data. The HTTPS connector currently cannot pass data as JSON content, only as form-encoded data.

This connector is particularly useful in two different cases:

1. You need to use a third-party service that normally takes its inputs from a web form. Perhaps you're replacing the form and need to maintain the existing process, or maybe you're just adding another process to your workflow. FormAssembly will act as the "middleman," processing your form and delivering all its advanced features, and then forwarding the data to the remote script.
2. You need to enforce some sort of validation on your form data that isn't provided by FormAssembly. Using a custom-built script, you could take the submitted data, check it against your own validation rules, and if needed, prevent the submission and redirect the respondent to the form with the appropriate error message.

**Note:** The HTTPS connector **cannot** be used for processing payments or sending sensitive data through the connector. Instead, please use one of our [Payment Connectors](#). Additionally, we enforce modern SSL TLS 1.2/HTTPS protocols for the HTTPS connector.

When referencing the HTTPS connector within scripts, we require the use of best practices and the standard port: tcp/443

---

## Requirements

- FormAssembly Essentials plan and higher, or legacy Professional and Premier plans

---

## Remote Script & Authentication

Provide the address (URL) of the script that will process the data. If needed, you can also enter a username and password (RFC 2617 Basic HTTP Authentication).



The screenshot shows a configuration form titled "REMOTE SCRIPT". It contains three main sections: "REMOTE SCRIPT URL:" with a text input field and a small icon on the right; "HTTP REQUEST METHOD:" with a dropdown menu currently set to "POST (default)"; and a checkbox labeled "Use HTTP Authentication" which is currently unchecked.

The [HTTP Status code returned](#) determines the success of the remote script executed. At this time, HTTP 2xx codes (200 to 206) are considered successes, and HTTP 4xx and 5xx are considered failures. Other status codes, such as 3xx redirects, are ignored.

Finding the remote script URL will differ depending on where you are attempting to send data. In general, you'll want to visit the destination website (i.e. where you are trying to send the data), right-click on the form or data entry portion of that page, and click **view source**.

Next, you'll want to use your browser's **Find** command (usually ctrl+f or cmd+f) and search for "**action=**" without quotation marks.

Once you've found that, you'll want to copy the value that is after the equals sign and use that as your remote script URL.

---

## Field Mapping

The purpose of the field mapping process is to specify what information should be sent and how it should be identified.

Your form fields:	will be submitted as... (field name)
First Name	<input type="text" value="tfa_1"/>
Last Name	<input type="text" value="tfa_2"/>
I agree to whatever you're making me sign	<input type="text" value="tfa_3"/>

[▶ Show choice mapping](#)

Additional values	will be submitted as... (field name)
<input type="text"/>	<input type="text"/>

[+ Add another custom field](#)

The HTTPS Connector lists the fields present in your form with their default field name. You can edit the field names to match the names expected by your remote script. You can also enter additional fields with fixed values as needed.

Use of the **HTTP POST** option will submit your mapped data as if the data were from a webform containing the form fields defined in your mappings.

The **HTTP GET** option will submit your mapped data as URL-encoded key-value pairs such as `https://www.example.com?q=Johnson`, assuming your Remote Script URL was `https://www.example.com` and you had mapped the Last Name field to a field name of "q".

---

## Activity Log

The connector's activity log is particularly useful to troubleshoot configuration problems with your connector.

To access it, go to the Connectors page for the desired form and click on the log link for the HTTPS Connector.

---

## Create an Attachment from a Response

If you're connecting to a remote service that is designed to return a downloadable file, such as a PDF, the HTTPS Connector will automatically save this file and attach it to your response.

You may access the file using the aliases listed below in your FormAssembly configuration, wherever formulas/aliases are supported.

To use these aliases on the Thank You page or to pass them to Salesforce, the HTTPS Connector must be enabled.

### File Name

%%HTTP\_CONNECTOR\_ATTACHMENT%%

### Link to Download File

%%HTTP\_CONNECTOR\_ATTACHMENT\_URL%% (with FormAssembly login)

%%HTTP\_CONNECTOR\_ATTACHMENT\_UNPROTECTED\_URL%% (unprotected, without FormAssembly login)

---

## POST Response Alias

Submit Mode will create the formula alias `%%TFA_POST%%` containing the remote script's response.

The full content of the response (including any HTML markup) can be accessed with formulas using the alias `%%TFA_POST%%`.

---

## Video Tutorials

We have two example setups you can take a look at, and also some templates!

[GoToWebinar Integration](#)

[HTTPS Connector: MailChimp](#)

---