

In this article:

- [What are Webhooks?](#)
- [How does a Webhook integration compare to an API Integration?](#)
- [Why use Webhooks?](#)
- [How Ascent360 Webhook Data Collection works:](#)
- [Technical Details:](#)
- [Webhook Data Processing](#)

What are Webhooks?

Ascent360 uses Webhooks as one way of collecting data from client applications. Webhooks are very quickly becoming a popular way of exchanging data between two applications. Thousands of applications send data via webhooks which is a very easy and efficient method of data transmission. As an example, WooCommerce can send purchase transactions via Webhook. Whenever an order is created this data can be fired out to a URL of your choice. In the case of Ascent360, the URL would be <https://webhook.ascent360.com>. The Ascent360 URL, will catch the purchase data, parse it out and load it into our database.

How does a Webhook integration compare to an API Integration?

Webhooks are best understood as a comparison to a standard API Integration. The two are described below:

API: When Ascent360 integrates with an API, we will need to schedule a time to “check” the source system for new transactions. So, we may decide to check the source system every hour. This means that each hour, Ascent360 will go check to see if there are any new transactions. If a transaction occurred 59 minutes ago, our data will be 59 minutes old. We could go decide to check instead every 15 min or even every 5 min. This will create a load on our system as well as the client system—especially if there are no new transactions.

Webhook: With Webhook, the Ascent360 system will never go check for new transactions. When a transaction occurs, the Source system will simply post that transaction to our URL. This means we will largely get the transaction in real time and can then send an email based upon the transaction in real time.

Why use Webhooks?

Webhooks are used due to the ease of creating and integrating the webhook. Webhooks are usually setup in the client application and will send only the type of information that Ascent360 cares about. (orders, customers etc..)

How Ascent360 Webhook Data Collection works:

Ascent360 has a simple web application hosted in Microsoft cloud (Azure) which accepts HTTP Post messages to our designated end point. This endpoint is <https://webhook.ascent360.com/V1/Data/{GUID}>. Clients are

able to post XML or JSON packets to this URL. The basic data flow will work as follows:

1. Client Posts XML or JSON packet to the Acent360 Webhook URL.
2. Ascent360 will accept the XML or JSON packet only if it contains a client-specific GUID at the end of the URL for security purposes.
3. Ascent360 will store that raw packet in a secure location
4. Ascent360 will "parse" the data packet into component parts. The format of the packet must be defined in advance.
5. Ascent360 will push the parsed data into our database
6. Ascent360 will take some other action based upon the data that is defined in advance (such as send an email to the email address defined in the packet).

Technical Details:

Setup

Ascent360 will setup client data sources in advance. Within our system, we will define a specific data source, a client ID and a GUID. The GUID is for security.

The URL shared with the client is <https://webhook.ascent360.com/V1/Data/{GUID}>

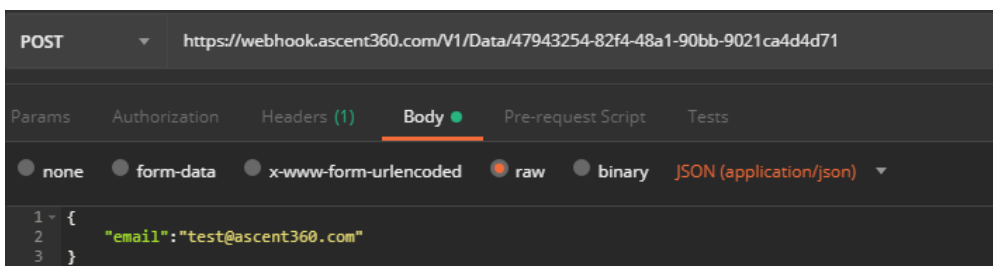
For example: <https://webhook.ascent360.com/V1/Data/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx>

Posting Data

We prefer JSON format but the data in other formats, such as XML can also be posted through HTTPS POST to the specific endpoint assigned.

Here is the Postman example:

1. UI



1. Code

```
var client = new RestClient("https://webhook.ascent360.com/V1/Data/{GUID});
```

```
var request = new RestRequest(Method.POST);
```

```
request.AddHeader("cache-control", "no-cache");
```

```
request.AddHeader("Content-Type", "application/json");
```

```
request.AddParameter("undefined", "{\n\t\"email\": \"test@ascent360.com\"\n}\n\n",  
ParameterType.RequestBody);
```

```
IRestResponse response = client.Execute(request);
```

Webhook Data Processing

We have different options to process webhook data received in real-time. These are generically defined below:

- Generic JSON Parser: We process the JSON and stage the data for CDP if JSON is simple
- Customized JSON parsing based on the requirement
 - If JSON is complex and specific data needs to be pulled and mapped to staging then a custom development can be done
- Custom format parsing:
 - If the client application will be sending some other format, we can parse this with advanced understanding of the requirements.