

Overview

This article contains documentation that outlines how clients can send contact data via webhook to Ascent360. This integration method does require some element of development time and should be discussed with your CSM or the help desk. **Note: On 12/2/2022 we added Source Type as a required field that must be present in webhook payloads.**

Webhook data can be transferred using server or client-side script. A JSON request body must be sent to the endpoint (webhook URL) in the specified format using the POST method.

In turn, if a provider and list ID(s) are specified, Ascent360 sends the contact data to the provider. Submissions are also loaded into the Customer Data Platform (Ascent360 marketing platform) and tagged with the Source Name submitted. Only a generic (200 / 4xx / 5xx) HTTP response is returned to the requester.

You will be provided with a custom webhook URL to use.

Parametres to Pass in the Request Body

Field	Include in Request Body As	Accepted Data Formats	Required Field?	Note
Source Name	SourceName	String	Required	Populates Source Name field in Ascent360.
Source Type	SourceType	String	Required	Populates Source Type field in Ascent360.
First Name	FirstName	String	Recommended	
Last Name	LastName	String	Optional	
Date of Birth	DateOfBirth	Date	Optional	mm-dd-yyyy
Gender	Gender	Text - "male" / "female"	Optional	Only "Male" and "Female" are stored. Other values display as "Unknown" in Ascent360.
Email	Email	Email (string)	Recommended	Email address.

Explicit Email Permission	EmailPermission	Boolean - true / false	Recommended	Denotes a change in explicit opt in / opt out from email communication. Please do not pass true / false by default. Exclude if explicit consent isn't given or withdrawn.
Phone	Phone	International phone (string)	Optional	Explicit Phone Permission is applied to this field.
Mobile	Mobile	international phone (string)	Optional	Mobile phone number.
Explicit Phone Permission	PhonePermission	Boolean - true / false	Optional	Denotes a change in explicit opt in / opt out for phone communication. Permissions are applied to the "Phone" field, not "Mobile". This field is not used for SMS / Mobile communications. Exclude if explicit consent isn't given or withdrawn.
Explicit SMS Permission	SMSPermission	Boolean - true / false	Optional	Denotes a change in explicit opt in / opt out for SMS communication. Permissions are applied to the "Phone" field, not "Mobile". Exclude if explicit consent isn't given or withdrawn.
Address	Address	JSON object		The address block must be passed within the address JSON object. See previous example.
ADDRESS LINE 1	Addressline1	String	Optional	Field in address block
ADDRESS LINE 2	Addressline2	String	Optional	Field in address block
CITY	City	String	Optional	Field in address block
STATE	State	String	Optional	Field in address block
ZIP	Zip	String	Optional	Field in address block

COUNTRY	Country	String	Optional	Field in address block
Explicit Address Permission	AddressPermission	Boolean - true / false	Optional	Denotes a change in explicit opt in / opt out from contact by mail. Exclude if explicit consent isn't given or withdrawn.
Activities	Activities	Array - text values	Optional	An array of text values. Please contact your CSM if you anticipate needing this field.
Interests	Interests	Array - text values	Optional	An array of text values. Please contact your CSM if you anticipate needing this field.
Newsletters	NewsLetters	Array - text values	Optional	An array of text values. Please contact your CSM if you anticipate needing this field.
Provider Type	ProviderType	numeric	Optional	This is the ID of the provider you want to send contacts to – i.e. Acoustic is "7". Mailchimp is "3". If the data <i>just</i> needs to flow into Ascent360 (and no other location) you do not need to include this field.
Provider List IDs	ListIds	Array - text / numeric values	Optional	This is an array of lists a form submitter should be added to <i>within</i> the provider (defined by provider ID). (i.e. ID of newsletter list).

If implementing in the web browser, create a JSON string using JavaScript then call the webhook via an XMLHttpRequest. Only the POST method will work. Alternately, this can be accomplished through server-side scripting (i.e. C#, Java, Rust, etc.).

```
var xhttp = new XMLHttpRequest();
xhttp.open("POST", "--insert custom webhook URL here-- ");
xhttp.send( --insert JSON string here-- );
```

Example JSON HTTP Request Body

```
{
  "SourceName": "Email Signup Form",
  "SourceType": "Webforms",
  "FirstName": "James",
  "LastName": "Smith",
```

```
"DateOfBirth": "12-01-1980",
"Gender": "Male",
"Email": "ABC@test.com",
"EmailPermission": "true",
"Phone": "19801234567",
"Mobile": "19801234567",
"PhonePermission": "false",
"SMSPermission": "true",
"Address": {
    "addressline1": "444 South",
    "Addressline2": "123 Maple Street",
    "city": "Pretendville",
    "state": "NY",
    "zip": "12345",
    "country": "USA"
},
"AddressPermission": "false",
}
```

Dos/Don'ts

- **Please strictly follow the format provided** since a valid JSON object will not produce an error in the response body. Even if the fields aren't correctly aliased, you will get a 200 HTTP response code if the JSON is valid.
- Values passed are not case-sensitive. Use the example request body for key-value pair naming.