

Table of Contents

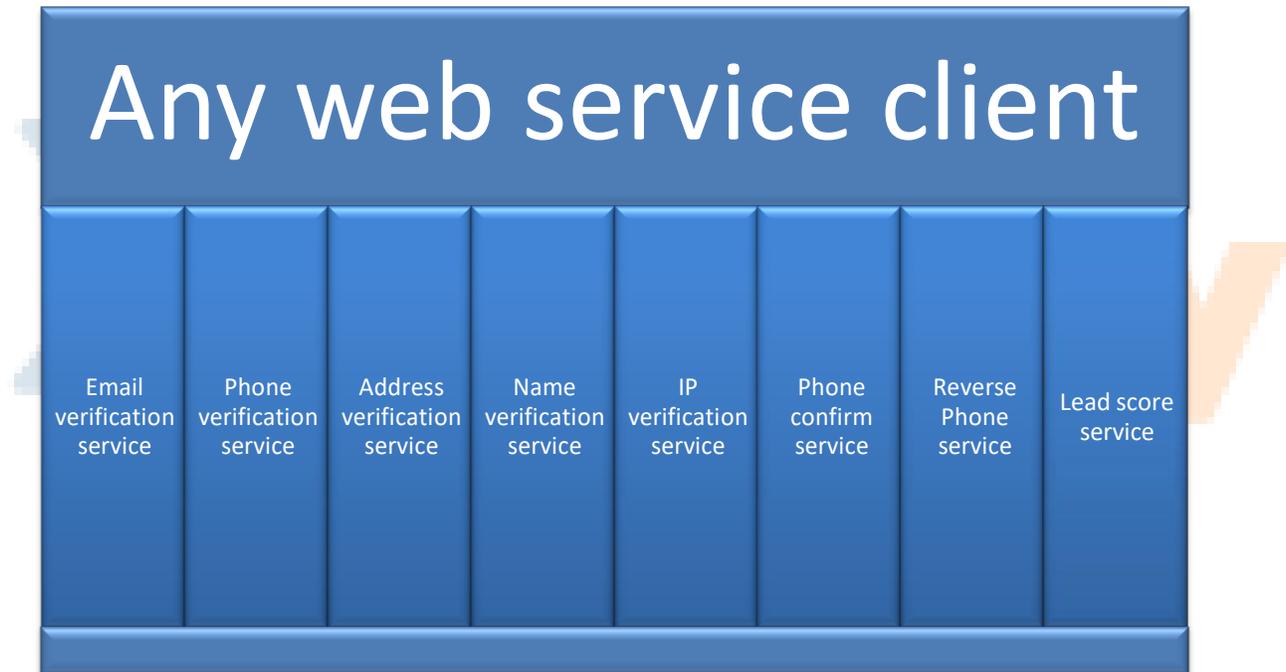
What is Xverify?	2
What is email verification service?	2
What is phone verification service?	3
What is address verification service?	4
What is name verification service?	4
What is IP verification service?	4
What is Phone Confirm verification service?	5
What is Reverse phone service?	5
What is Lead score service?	6
What is All in One API service?	6
What is Postback URL?	6
How to use Xverify?	7
Plug and play script:	7
Sample script:	7
Restful call:	14
Email verification Request/Response:.....	15
Phone verification Request/Response:.....	17
Address verification Request/Response:.....	18
Name verification Request/Response:.....	21
IP verification Request/Response:	22
Phone confirm verification Request/Response:	23
Reverse phone Request/Response:.....	24
Lead score Request/Response:.....	26
All in One API Request/Response:.....	28
Postback URL Request/Response:.....	29
Affiliate Stats API Request/Response:	31
Sub Affiliate Stats API Request/Response:.....	32
Add Domain API Request/Response:	34
Add Postback Job Request/Response:	34
Get Postback Job Status Request/Response:.....	35
Get Postback Job Detail Request/Response:	36

What is Xverify?

Xverify is mainly a verification platform which offers a bundle of verification services. The list is given below:

1. Email verification service
2. Phone verification service
3. Address verification service
4. Name verification service
5. IP verification service
6. Phone confirm service
7. Reverse phone service

If we have a look over the abstract level, the following diagram can help us a lot in understanding its architecture:



Here the web service client can be any entity capable of generating Restful call over HTTP transport layer. The system has a great flexibility to return the response in either XML or in JSON. That's why; the calling party (i.e. web service client) should be able to parse either of them.

Let's go through of each service and its details.

What is email verification service?

Email verification service is a mechanism by which we ensure that the email is valid or not. By saying valid or not, we mean that it actually exists in this World or not. As an input, it takes an email address from the user and sends the response with the following information:

1. Address (user provided email address)
2. Auto Correct (represents the auto correct information if performed)
3. Syntax (represents whether the syntax is good or not)
4. Handle (represents the username part of the email)
5. Domain (represents the domain name part of the email)
6. Response Code (ranging from 1 to 12)
7. Error (if non zero, means error occurred)
8. Status (either "valid" or "invalid")
9. Duration (the time it takes to verify)
10. Message (represents message to be prompted in case of error)
11. Catch All (represents whether the domain associated with the email is catch all or not)

What is phone verification service?

In the phone verification service, there are further 2 sub categories on which this service can be dig down:

1. Cell phone verification
2. Landline verification

In all two sub categories, we provide the user whether the provided number is valid or not based on user's subscriptions and in the response; we give the following information back:

1. Value (user provided phone number)
2. Error (if non zero, means error occurred)
3. Status (either "valid" or "invalid")
4. Duration (the time it takes to verify)
5. Message (represents message to be prompted in case of error)
6. Prefix (represents the provided number prefix)
7. Suffix (represents the provided number suffix)
8. Latitude (represents the latitude based on the provided number)
9. Longitude (represents the longitude based on the provided number)
10. Extension (represents the extension associated with the provided number if any)
11. City (represents the relevant city based on the provided number)
12. State (represents the relevant state based on the provided number)
13. Country (represents the relevant country based on the provided number)
14. Time zone (This tag has two children tags)
 - a. Code (represents the relevant time zone code in integer based on the provided number)
 - b. Name (represents the relevant time zone name based on the provided number)
15. Exchange Type (represents the provided number exchange type. E.g. landline, cell, voip etc)
16. Phone Type (represents the provided number phone type. E.g. business, residential etc)

Note:

1. *The user can subscribe to any one or all of the phone services based on his business needs.*

What is address verification service?

In the address verification service, we perform verification check to let the user know about the address validity. If any address is valid, that means it can be reached by any Postal service. In this service, we take the following values from the user:

1. Address (Mandatory)
2. City (Optional)
3. State (Optional)
4. Zip (Mandatory)

And performs address validation based on the provided values. It also auto corrects the address if some mistake is made by the user. In the response, it gives the following information:

1. Error (if non zero, means error occurred)
2. Status (either “valid” or “invalid”)
3. Reason for Invalid (contains description if available incase the address is detected as invalid)
4. Message (represents message to be prompted in case of error)
5. Duration (the time it takes to verify)
6. Street (the provided address from the user)
7. Zip (the provided zip from the user)
8. City (represents the city. This input field is optional and if leave empty by the user, the XVerify system fills the relevant city automatically based on the address and zip)
9. State (represents the state. This input field is optional and if leave empty by the user, the XVerify system fills the relevant state automatically based on the address and zip)

What is name verification service?

Name verification service is a way of verifying the names to let the user know whether the name is valid or not. Furthermore, on top of it, it also executes the profanity filter on it and prompts if detected. It takes the following values from the user:

1. First Name
2. Last Name

And based on that, either verifies the name or prompts the user about its status. In the response, it gives the following information:

1. Value (user provided input)
2. Status (either “valid” or “invalid”)
3. Duration (the time it takes to verify)
4. First (represents first name)
5. Last (represents last name)
6. Error (if non zero, means error occurred)
7. Message (represents message to be prompted in case of error)

What is IP verification service?

IP verification service is a great way to know about its validity. It takes the IP as input and if it's verified, gives the following information back to the user:

1. Region
2. City
3. Country
4. Latitude
5. Longitude

What is Phone Confirm verification service?

Phone confirm service is really a powerful service to verify the phone numbers based on human interaction. It speaks a code in the call that the user is supposed to enter in the intended application. Every placed call corresponds to a unique transaction number. That transaction number is queried later on to verify whether the user has entered the code correctly or not. Based on this verification, the intended system allows or rejects his access. Please note that this service is only available through data append. It's not available from JavaScript.

What is Reverse phone service?

Reverse phone service is another great service which works in opposite direction. That means it takes phone number as input and looks up for his information bind against his provided phone number. It provides the following information back to the user:

1. Name (represents text of the listing name, both first and last, or the business name)
2. First Name (represents the first name of the residential listing (or the name of both people at a. The address, i.e. ("John & Mary"). This field is not populated for business listings.)
3. Last Name (represents the last name of the residential listing. This field is not populated for business listings.)
4. Full Address (represents the full address of the person)
5. House Number (represents the text of the house number)
6. House Fraction (represents the fractional portion of the house number. I.E. "1/2", or "B".)
7. Pre directional (represents the directional that is placed in front of the street name; "SE", or "NW")
8. Street (represents the name of the street that the address is on)
9. Street Type (represents the type of street that it is, i.e. Ave or Blvd)
10. Post directional (represents directional after the name of the street; "NW" or "SE".)
11. Suit No (represents the suite, PO Box number or apartment number of the address)
12. Latitude (represents the latitude, in degrees (i.e. 45.12345).)
13. Longitude (represents the longitude, in degrees (i.e. -120.56789).)
14. City (represents the city of the listing.)
15. State (represents the state of the listing)
16. Zip5 (represents the first five digits of zip code)
17. ZIP4 (represents the first four digits of zip code)
18. NPA (represents the area code of listing phone number.)
19. NXX (represents the exchange of the listing phone number.)
20. Postfix (represents the suffix of the listing phone number.)
21. Listing Type (represents business, residential or government with "B", "R", or "G" respectively.)
22. Flags (represents different flags associated with the listing, such as non-pub or non-listed)

23. Error (if non zero, means error occurred)
24. Status (either “valid” or “invalid”)

What is Lead score service?

Lead score service is a service that is used to match the user provided information with the actual information. Based on the amount of matched information, we provide results in terms of percentage. Percentage represents the amount of matched information out of the total information provided to the system. It takes the following information from the user:

1. Phone Number
2. First Name
3. Last Name
4. Address
5. City
6. State
7. Zip
8. Email
9. IP

Based on this information provided by the user, we first perform reverse phone lookup to find out its actual information. Based on this actual information, we make comparison and along the percentage, also return what is matched and what is not? A part of this, we also inform the user whether the provided email, phone and address is validated or not.

What is All in One API service?

All in One API is another convenient way of verifying all of the services in just a single call. What this mean is that if you want to verify email, address and phone, you don't need to make separate calls but you can place one call to our system providing all of the information and our system will do rest of his magic. For now, we are covering the following three services in All in One API:

1. Email
2. Phone
3. Address

What is Postback URL?

There are several situations in which we need to submit our verification results to some third party. This is required when our system is integrated with that third party for some purpose utilizing the verification results. We can provide our information as well as use Xverify service verification results to be forwarded to our concerned third party. Postback URL is available for all of the available services we have in the Xverify system.

How to use Xverify?

There are three methods that you can use to use XVerify:

1. Plug and play script
2. Sample script
3. Restful call

Note:

1. *The following methods assume that you are already registered with www.xverify.com*

Plug and play script:

In the first method, you get a readymade script that is supposed to be uploaded over the web server to feel the experience in just a few minutes. If you want to use this method, the following guideline will walk you through it:

1. Please use your credentials to login to the system (i.e. www.xverify.com)
2. Please click on the “Implementation” tab.
3. In the Implementation page, you will see a link named as “Download Sample”, please click on that link. The web browser will start downloading the HTML file.
4. Once the HTML file is downloaded to your machine, please upload it to any available web server.
5. Once you are done with the uploading, please hit the desired URL in any web browser and you are good to go!

Note:

1. *The intention of providing this script is to show you the services in actions with minimum effort. You are not recommended to use the source code of this script to integrate your systems with XVerify. For integrations, please read the immediate following method (i.e. Sample script)*
2. *In the current and the immediate following method, the request is generated by JavaScript and response is received in JSON.*

Sample script:

In this method, you can integrate your system with XVerify. Here you get a JavaScript code that you are supposed to embed in your presentation layer under the HTML head tag. If you take a look in the source code of the plug and play script, you will notice that under the head tag, the same JavaScript code is embedded which you got in this method. So, the source code of the sample plug and play script can help you a lot in integrating Xverify services in your existing systems.

Note:

1. Please include JavaScript file named as “**jquery.xverify.plugin.js**” which is compatible with all of the following properties:
 - a. “class” property
 - b. “id” property
 - c. “name” property
2. What does it mean actually is that if any one or more of the above mentioned properties are available in the HTML input elements, our script will work.
3. Please note that you should include only one JavaScript file in the same HTML page

Following script is use to initialize the javascript on page now

```
$.xVerifyService({
  services: {
    email: {field: 'field_name|field_id|field_class'},
    phone: { field: 'field_name|field_id|field_class'},
    address: {
      street_field: 'field_name|field_id|field_class',
      street_field_2: 'field_name|field_id|field_class',
      zip_field: 'field_name|field_id|field_class'
    }
  },
  submitType: 'onChange | Onsubmit',
  formButtons: ['xvButton'],
});
```

All parameters are optional here. You can also define the array of fields in services on following way

```
$.xVerifyService({
  services: {
    email: {fields: ['emailField', 'emailField_1', 'emailField_2']},
    phone: { fields: ['phoneField', 'phoneField_1', 'phoneField_2']},
    address: {
      fields: [
        {street: 'xvStreet',street_2: 'xvStrret2',zip: 'xvXip'},
        {street: 'xvStreet',street_2: 'xvStrret2',zip: 'xvXip'},
      ]
    }
  },
  submitType: 'onChange | Onsubmit',
  formButtons: ['xvButton_1', 'xvButton_2', 'xvButton_3'],
});
```

For more details please check the following details

The “**services**” property is use to customize the input field in name/id/class for each service now. You can specify your own field attribute when initialize the JavaScript. When no properties define on initialize time then its use default properties now. Default its use the following properties for email services Id=email or name=email or class=xverify_email.

You can also define list of array in fields attribute then its bind the xverify on anyone of the field found on the page now.

Default attributes for each service describe below table with name “**Rule if you are subscribed for verification service**”

The “**submitType**” property in the javascript use to check the service on which event on web page if you not specified then its use the default onChange event but we can fire on form submit also

The “**formButtons**” property in the javascript use to disable the button on form if you not specified then its disable the submit button on the form now. When you define list of buttons on script then its disable all button which define in script and exist on the page now

The following guideline will walk you through it:

1. Please use your credentials to login to the system (i.e. www.xverify.com)
2. Please click on the “Implementation” tab.
3. In the Implementation page, you will see a link named as “Download script”, please click on that link. The web browser will start downloading the Text file.
4. Once the Text file is downloaded to your machine, please open it and copy the JavaScript Code in your desired file where you want to integrate it. This file includes the latest JavaScript file which is just added into the system to give you the maximum flexibility and control in your coding. But please make sure you follow the following rules mentioned in the following table

Note:

1. Please note that now we support phone verification service in three fields format also. That means, user can provide 3 digits in the first two text fields and 4 digits in the third text field and our system will verify it.

Rule if you are subscribed for email verification service	
Rule	Example
The “class” property of the email text field should be named as “xverify_email”	<input class =“xverify_email” type=“text”>
OR	
The “id” property of the email text field should be named as “email”	<input id =“email” type=“text”>
OR	
The “name” property of the email text field should be named as “email”	<input name =“email” type=“text” >
OR	

You can have all of them in one element	<code><input id="email" name="email" class="xverify_email" type="text"></code>
You can have all of them in one element	<code><input id="email" name="email" class="xverify_email" type="text"></code>
Rule if you are subscribed for Landline verification service	
Rule	Example
The “class” property of the landline text field should be named as “xverify_landline”	<code><input class="xverify_landline" type="text" ></code>
OR	
The “id” property of the landline text field should be named as “landline”	<code><input id="landline" type="text" ></code>
OR	
The “name” property of the landline text field should be named as “landline”	<code><input name="landline" type="text" ></code>
OR	
You can have all of them in one element	<code><input id="landline" name="landline" class="xverify_landline" type="text" ></code>
Rule if you are subscribed for multi fields Landline verification service	
Rule	Example
The “class” property of the multi field landline text fields should be named as “xverify_multifield_landline_1”, “xverify_multifield_landline_2” and “xverify_multifield_landline_3” respectively.	<code><input class="xverify_multifield_landline_1" type="text" ></code> <code><input class="xverify_multifield_landline_2" type="text" ></code> <code><input class="xverify_multifield_landline_3" type="text" ></code>
OR	
The “id” property of the multi field landline text fields should be named as “multifield_landline_1”, “multifield_landline_2” and “multifield_landline_3” respectively.	<code><input id="multifield_landline_1" type="text" ></code> <code><input id="multifield_landline_2" type="text" ></code> <code><input id="multifield_landline_3" type="text" ></code>
OR	
The “name” property of the multi field landline text fields should be named as “multifield_landline_1”, “multifield_landline_2” and “multifield_landline_3” respectively.	<code><input name="multifield_landline_1" type="text" ></code> <code><input name="multifield_landline_2" type="text" ></code> <code><input name="multifield_landline_3" type="text" ></code>
OR	
You can have all of them in one element	<code><input class="xverify_multifield_landline_1" id="multifield_landline_1" name="multifield_landline_1" type="text" ></code> <code><input class="xverify_multifield_landline_2" id="multifield_landline_2" name="multifield_landline_2" type="text" ></code> <code><input class="xverify_multifield_landline_3" id="multifield_landline_3" name="multifield_landline_3" type="text" ></code>

Note:

Please note that "multifield_landline_1 or xverify_multifield_landline_1" should be set to first landline text field, "multifield_landline_2 or xverify_multifield_landline_2" should be set to second landline text field and "multifield_landline_3 or xverify_multifield_landline_3" should be set to the third landline text field.

Rule if you are subscribed for Cell verification service

The "class" property of the cell text field should be named as "xverify_cell"

```
<input class="xverify_cell" type="text">
```

OR

The "id" property of the cell text field should be named as "cell"

```
<input id="cell" type="text">
```

OR

The "name" property of the cell text field should be named as "cell"

```
<input name="cell" type="text">
```

OR

You can have all of them in one element

```
<input id="cell" name="cell" class="xverify_cell" type="text">
```

Rule if you are subscribed for multi fields Cell verification service**Rule****Example**

The "class" property of the multi field landline text fields should be named as "xverify_multifield_cell_1", "xverify_multifield_cell_2" and "xverify_multifield_cell_3" respectively.

```
<input class="xverify_multifield_cell_1" type="text" >
<input class="xverify_multifield_cell_2" type="text" >
<input class="xverify_multifield_cell_3" type="text" >
```

OR

The "id" property of the multi field landline text fields should be named as "multifield_cell_1", "multifield_cell_2" and "multifield_cell_3" respectively.

```
<input id="multifield_cell_1" type="text" >
<input id="multifield_cell_2" type="text" >
<input id="multifield_cell_3" type="text" >
```

OR

The "name" property of the multi field landline text fields should be named as "multifield_cell_1", "multifield_cell_2" and "multifield_cell_3" respectively.

```
<input name="multifield_cell_1" type="text" >
<input name="multifield_cell_2" type="text" >
<input name="multifield_cell_3" type="text" >
```

OR

You can have all of them in one element

```
<input class="xverify_multifield_cell_1"
id="multifield_cell_1"
name="multifield_cell_1"
type="text" >
<input
class="xverify_multifield_cell_2" id="multifield_cell_2"
name="multifield_cell_2" type="text" >
<input class="xverify_multifield_cell_3"
id="multifield_cell_3"
name="multifield_cell_3" type="text" >
```

Note:

Please note that "multifield_cell_1 or xverify_multifield_cell_1" should be set to first cell text field, "multifield_cell_2 or xverify_multifield_cell_2" should be set to second cell text field and "multifield_cell_3 or xverify_multifield_cell_3" should be set to third cell text field.

_3 or xverify_multifield_cell_3"should be set to the third cell text field.	
Rule if you are subscribed for all kind of Phone verification services (i.e. Landline and Cell)	
Rule	Example
The "class" property of the phone text field should be named as "xverify_phone"	<input class="xverify_phone" type="text">
OR	
The "id" property of the phone text field should be named as "phone"	<input id="phone" type="text">
OR	
The "name" property of the phone text field should be named as "phone"	<input name="phone" type="text">
OR	
You can have all of them in one element	<input id="phone" name="phone" class="xverify_phone" type="text">
Rule if you are subscribed for all kind of multi fields Phone verification services (i.e. Landline and Cell)	
Rule	Example
The "class" property of the multi field landline text fields should be named as "xverify_multifield_phone_1", "xverify_multifield_phone_2" and "xverify_multifield_phone_3" respectively.	<input class="xverify_multifield_phone_1" type="text" > <input class="xverify_multifield_phone_2" type="text" > <input class="xverify_multifield_phone_3" type="text" >
OR	
The "id" property of the multi field landline text fields should be named as "multifield_phone_1", "multifield_phone_2" and "multifield_phone_3" respectively.	<input id="multifield_phone_1" type="text" > <input id="multifield_phone_2" type="text" > <input id="multifield_phone_3" type="text" >
OR	
The "name" property of the multi field landline text fields should be named as "multifield_phone_1", "multifield_phone_2" and "multifield_phone_3" respectively.	<input name="multifield_phone_1" type="text" > <input name="multifield_phone_2" type="text" > <input name="multifield_phone_3" type="text" >
OR	
You can have all of them in one element	<input class="xverify_multifield_phone_1" id="multifield_phone_1" name="multifield_phone_1" type="text" > <input class="xverify_multifield_phone_2" id="multifield_phone_2" name="multifield_phone_2" type="text" > <input class="xverify_multifield_phone_3" id="multifield_phone_3" name="multifield_phone_3" type="text" >
Note:	

Please note that "multifield_phone_1 or xverify_multifield_phone_1" should be set to first phone text field, "multifield_phone_2 or xverify_multifield_phone_2" should be set to second phone text field and "multifield_phone_3 or xverify_multifield_phone_3" should be set to the third phone text field.

Rule if you are subscribed for Address verification service

	Example
The properties "id" and "name" should be named as "street" and "class" property should be named as "xverify_street"	<input id="street" type="text">
	OR
	<input name="street" type="text">
	OR
	<input class="xverify_street" type="text">
The properties "id" and "name" should be named as "city" and "class" property should be named as "xverify_city"	<input id="city" type="text">
	OR
	<input name="city" type="text">
	OR
	<input id="city" name="city" class="xverify_city" type="text">
The properties "id" and "name" should be named as "state" and "class" property should be named as "xverify_state"	<input id="state" type="text">
	OR
	<input name="state" type="text">
	OR
	<input id="state" name="state" class="xverify_state" type="text">
The properties "id" and "name" should be named as "zip" and "class" property should be named as "xverify_zip"	<input id="zip" type="text">
	OR
	<input name="zip" type="text">
	OR
	<input id="zip" name="zip" class="xverify_zip" type="text">

Rule if you are subscribed for Name verification service

Rule	Example
The properties "id" and "name" should be named as "firstname" and "class" property should be named as "xverify_firstname"	<input id="firstname" type="text">
	OR
	<input name="firstname" type="text">
	OR
	<input id="firstname" name="firstname" class="xverify_firstname" type="text">

The properties "id" and "name" should be named as "lastname" and "class" property should be named as "xverify_lastname"	<code><input id="lastname" type="text"></code>
	OR
	<code><input name="lastname" type="text"></code>
	OR
	<code><input class="xverify_lastname" type="text"></code>
OR	
	<code><input id="lastname" name="lastname" class="xverify_lastname" type="text"></code>
Rule if you are subscribed for IP verification service	
Rule	Example
The properties "id" and "name" should be named as "ipverify" and "class" property should be named as "xverify_ipverify"	<code><input id="ipverify" type="text"></code>
	OR
	<code><input name="ipverify" type="text"></code>
	OR
	<code><input class="xverify_ipverify" type="text"></code>
OR	
	<code><input id="ipverify" name="ipverify" class="xverify_ipverify" type="text"></code>

Note:

1. Please make sure you have encapsulated the desired fields in the HTML form tag. For more, please see "Plug and play script".

Affiliate and sub affiliate support:

We have affiliate and sub affiliate support in the system. In the web form, user can provide affiliate value in the parameter named as "v1" and sub affiliate value in the parameter named as "v2". These two parameters can be provided in either of the following two ways:

1. In the browser URL
2. In the hidden DOM elements

Please note that the ones provided in the browser will be given more precedence. If any information is missing in either way, it will be combined. So, for example: if v1 is given in browser URL but not in hidden DOM element. And v2 is given in the hidden DOM element but not in the browser URL, then our system will combine them both together and send them in our server.

Restful call:

The last but not the least method of integrating the XVerify services with your system is to call the web service directly. The advantage of directly calling the restful API is more flexibility in terms of control. Meaning that we can write our own business logic based on the response we get from the requested API call. Furthermore, we get responses in the following two notations:

1. XML
2. JSON

So, we can inform the notation type to XVerify when we make a call to API.

Based on service subscriptions, we get different URLs and every API call is based on HTTP GET method. Let's walk through of each service to know its required fields to call and the response fields as well.

Email verification Request/Response:

Here is the sample email verification service URL:

```
http://www.xverify.com/services/emails/verify/?email=<email>&type=<xml/json>&apikey=<api key>&domain=<allowed domain>
```

A part of this, now we also support JSON format as POST Payload with the same parameters.

Let's understand each parameter in the query string of the above provided sample email verification URL through the following table:

Request for email verification service	
Query String	Description
email (required)	This parameter contains the email address for which the user wants to verify
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit.
secret_key (optional)	This optional parameter if provided is used to automatically add a domain if not exist already.
Response for email verification service	
Response field	Description
address	This response field contains the same email address the user provided in the request.
syntax	This response field shows whether the provided email address syntax is good or not. It's possible values are 1 and 0. 1 means syntax is good and 0 means syntax is bad
handle	This response field contains the username part of the provided email address
domain	This response field contains the domain name part of the provided email address
error	This field indicates error. If it's non-zero, that means error occurred. You

		can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means the system is under maintenance and will be available soon.
status		This response field contains either “valid” or “invalid”
auto_correct		This response field contains the auto correct information, if auto correct feature is disabled in the email service setting page then auto_correct response tag remains empty otherwise it contains the following fields
	corrected	This response field is a child tag of “auto_correct” tag and contains either “true” or “false”. True means correction is performed over the email address which is explained below.
	address	If the tag “corrected” contains “true”, then corrected email address is occupied by this tag.
message		This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
responsecode		This field represents the specific response types. Here are the codes and their respective meanings: Response Code 1 – Valid Email Address Response Code 2 – Email Address Does Not Exist Response Code 3 – Unknown Response Code 4 – Fraud List Response Code 5 – High Risk Email Address Response Code 6 – Affiliate Is Blocked By Client Response Code 7 – Complainer Email Address Response Code 8 – Top Level Domain Blocked By Client Response Code 9 – Temporary/Disposable Email Response Code 10 – Keyword is Blocked By Client Response Code 11 – IP address – Country Not Allowed Response Code 12 – Block list from Client Settings Response Code 400 – Missing required fields Response Code 503 – Invalid API Key/Service Not Active Response Code 504 – User reach the API Limit
duration		This field indicates the total execution time for the request.
catch_all		This field indicates whether the domain associated with the email address is catch all or not. It's possible values are "Yes/No/Unknown". Please note that "Unknown" is returned when we get unable to find it's catch ability.

Note:

1. We can get this sample email verification URL from the Xverify back end system in the email settings page.
2. Test transactions count is specifically defined for development and testing purposes only.

Phone verification Request/Response:

Here is the sample phone verification service URL:

```
http://www.xverify.com/services/phone/verify/?phone=<phone>&type=<xml/json>
&apikey=<a pi key>&domain=<allowed domain>&phonetype=<landline/cell>
```

We can get this sample email verification URL from the XVerify back end system in the phone settings page.

Let's go through of each request and response fields one by one through the following table:

Request for phone verification service	
Parameter in the Query String	Description
Phone (required)	This parameter contains the phone for which the user wants to verify
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
phonetype (required)	This request field contains "cell" if the user wants to verify cell number, "landline" if he wants to verify landline number and "phone" if he wants to verify any one of them.
secret_key (optional)	This optional parameter if provided is used to automatically add a domain if not exist already.
Response for phone verification service	
Response field	Description
value	This response field contains the same phone number the user provided in the request.
error	This field indicates error. If it's non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means the system is under maintainance and will be available soon.
status	This response field contains either "valid" or "invalid"
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
duration	This is the time system takes to respond
areacode	This response field represents the relevant area code based on the provided phone number. For example, for the number "8472215200", the area code value is "847"

prefix	This is the number prefix. For example, for the number “8472215200”, the prefix value is “221”	
suffix	This is the number suffix. For example, for the number “8472215200”, the suffix value is “5200”	
latitude	This response field represents the relevant latitude based on the provided phone number	
longitude	This response field represents the relevant longitude based on the provided phone number	
extension	This response field represents the extension associated with the provided number if any	
city	This response field represents the relevant city based on the provided phone number	
state	This response field represents the relevant state based on the provided phone number	
exchange_type	This response field represents the provided phone number exchange type. For example, “landline” for landline number, “cell” for cell phone number and “voip” for voip number.	
response_code	<p>This field represents the specific response types. Here are the codes and their respective meanings:</p> <p>Response Code 1 – Valid Phone Response Code 2 – Phone Does Not Exist Response Code 3 – Unknown Response Code 400 – Missing required fields Response Code 503 – Invalid API Key/Service Not Active Response Code 504 – User reach the API Limit</p>	
phone_type	This response field represents the provided phone number type. For example, “business” for business numbers, “residential” for residential numbers and “soho” for small business or home office.	
Country		This response field represents the relevant country based on the provided phone number. This response field contains two sub children response fields explained below.
	abbreviation	This child response field represents the short code for the country.
	name	This child response field represents the complete name of the country.
time zone	code	This response field represents the relevant time zone code in integer based on the provided number
	name	This response field represents the relevant time zone name based on the provided number

Address verification Request/Response:

Here is the sample address verification service URL:

```
http://www.xverify.com/services/address/verify/?street=<street address> &zip=<zip code>&type=<xml/json>&apikey=<apikey>&domain=<allowed domain>
```

We can get this sample address verification URL from the XVerify back end system in the address settings page.

Let's go through of each request and response fields one by one through the following table:

Request for address verification service	
Parameter in the Query String	Description
street (required)	This parameter contains the address for which the user wants to verify
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
city (optional)	This is an optional field that can be sent in the request. If however, not mentioned in the request, the system will send back the suggested city in the response.
state (Optional)	This is an optional field that can be sent in the request. If however, not mentioned in the request, the system will send back the suggested state in the response.
secret_key (optional)	This optional parameter if provided is used to automatically add a domain if not exist already.
Response for address verification service	
street	This response field contains the same address the user provided in the request.
error	This field indicates error. If it's non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means the system is under maintainance and will be available soon.
status	This response field contains either "valid" or "invalid"
reason_for_invalid	This response field contains the description, if available, for invalid address for the address elements which could not be found.
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
duration	This is the time system takes to respond.
city	This response field is the relevant city based on the provided street address and zip.
state	This response field is the relevant state based on the provided street address and zip.
zip	This response field contains the same zip the user provided in the request.

response_code	<p>This field represents the specific response types. Here are the codes and their respective meanings:</p> <p>responsecode = 0 means valid address</p> <p>responsecode = 1 The Postal Code does not exist and could not be determined by the city/municipality and state/province.</p> <p>responsecode = 2 An exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.</p> <p>responsecode = 3 Either the directionals or the suffix field did not match the post office database, or there was more than one choice for correcting the address.</p> <p>responsecode = 4 The physical location exists but there are no homes on this street. One reason might be railroad tracks or rivers running alongside this street, as they would prevent construction of homes in this location.</p> <p>responsecode = 5 Address matched to multiple records. More than one record matches the address and there is not enough information available in the input address to break the tie between multiple records.</p> <p>responsecode = 6 This address has been identified in the Early Warning System (EWS) data file and should be included in the next postal database update.</p> <p>responsecode = 7 Minimum required input of address/city/state or address/zip not found.</p> <p>responsecode = 8 The input street address was found but the input suite number was not valid.</p> <p>responsecode = 9 The input street address was found but a required suite number is missing.</p> <p>responsecode = 10 The street number in the input address was not valid.</p> <p>responsecode = 11 The street number in the input address was missing.</p> <p>responsecode = 12 The input address PO, RR or HC number was invalid.</p> <p>responsecode = 13 The input address is missing a PO, RR, or HC Box number.</p> <p>responsecode = 14 Address Matched to a CMRA Address but the secondary (Private mailbox number) is missing.</p> <p>responsecode = 17 A suite number was entered but no suite information found for primary address.</p> <p>responsecode = 20</p>
----------------------	---

	PO BOX not accepted responsecode = 21 means IP address – Country Not Allowed responsecode = 400 means missing required fields responsecode = 503 means invalid API key/service not active responsecode = 504 means user reach the API Limit
--	---

Name verification Request/Response:

Here is the sample name verification service URL:

```
http://www.xverify.com/services/name/verify/?firstname=<first name>&lastname=<last name>&type=<xml/json>&apikey=<api key>&domain=<allowed domain>
```

We can get this sample address verification URL from the XVerify back end system in the name/IP settings page.

Let’s go through of each request and response fields one by one through the following table:

Request for name verification service	
Parameter in the Query String	Description
Firstname (required)	This request parameter contains the first name for which the user wants to verify
Lastname (required)	This request parameter contains the last name for which the user wants to verify
Apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
Domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won’t entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
Type (required)	This parameter can take either “xml” or “json”. If we want to get the response in XML, then we need to pass “xml”, otherwise, we need to pass “json”
secret_key(optional)	This optional parameter if provided is used to automatically add a domain if not exist already.
Response for name verification service	
Response field	Description
value	This response field contains the same name the user provided in the request.
error	This field indicates error. If it’s non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means

	the system is under maintainance and will be available soon.
status	This response field contains either “valid” or “invalid”
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
duration	This is the time system takes to respond
first	This response field is the same first name the user provided in the request.
last	This response field is the same last name the user provided in the request.
response_code	This field represents the specific response types. Here are the codes and their respective meanings: Response Code 1 – Valid Name Response Code 2 – Invalid Name Response Code 400 – Missing required fields Response Code 503 – Invalid API Key/Service Not Active

IP verification Request/Response:

Here is the sample IP verification service URL:

```
http://www.xverify.com/services/ipdata/verify/?ip=<ip>&type=<xml/json>&apikey=<api key>&domain=<allowed domain>
```

We can get this sample address verification URL from the XVerify back end system in the name/IP settings page.

Let’s go through of each request and response fields one by one through the following table:

Request for IP verification service	
Parameter in the Query String	Description
ip (required)	This request parameter contains the IP for which the user wants to verify
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won’t entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
type (required)	This parameter can take either “xml” or “json”. If we want to get the response in XML, then we need to pass “xml”, otherwise, we need to pass “json”
secret_key (optional)	This optional parameter if provided is used to automatically add a domain if not exist already.
Response for IP verification service	

	This response field contains the same name the user provided in the request.
error	This field indicates error. If it's non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means the system is under maintenance and will be available soon.
status	This response field contains either "valid" or "invalid"
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
duration	This is the time system takes to respond
region	This response field represents the relevant region based on the IP provided in the request
city	This response field represents the relevant city based on the IP provided in the request
country	This response field represents the relevant country based on the IP provided in the request
latitude	This response field represents the relevant latitude based on the IP provided in the request
longitude	This response field represents the relevant longitude based on the IP provided in the request
response_code	This field represents the specific response types. Here are the codes and their respective meanings: Response Code 1 – Valid IP address Response Code 2 – Invalid IP address Response Code 400 – Missing required fields Response Code 503 – Invalid API Key/Service Not Active

Phone confirm verification Request/Response:

Here is the sample Phone confirms verification service URL:

```
http://www.xverify.com/services/phoneconfirm/placecall/?phone=<phone_number>&country_code=<country_code>&code=<code_to_speak>&type=<xml/json>&&apikey=<api_key>&domain=<allowed_domain>
```

We can get this sample address verification URL from the XVerify back end system in the Phone confirm settings page.

Let's go through of each request and response fields one by one through the following table:

Request for Phone confirm verification service	
Parameter in the Query String	Description
phone (required)	This request parameter contains the phone number where the call will be placed

country_code (required)	This request parameter contains the country code where the call will be placed
code (required)	This request parameter contains the code to speak during the placed call
redial_count (optional)	This request parameter is used to redial the call in case it's not picked for the
redial_interval (optional)	This request parameter is used to define the interval between the redial attempts. It's necessary to provide with the redial_count parameter. It's given in seconds.
call_place_time(optional)	This request parameter contains the time to place the call for the first time. It's
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
type (required)	This parameter can take either "xml" or "json". If we want to get the response
secret_key (optional)	This optional parameter if provided is used to automatically add a domain if

Response for Phone confirm verification service

Response field	Description
transaction_number	This response field contains the transaction number which uniquely identifies
error	This field indicates error. If it's non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have - 1 value, that means the system is under maintainance and will be available soon.
Message	This response field contains the actual message in human readable form.

```
http://www.xverify.com/services/
reversephone/verify/?phone=<phone_number>&type=<xml/json>&&apikey=<api_key
>&domain
=<allowed_domain>
```

Reverse phone Request/Response:

Here is the sample Reverse phone service URL:

We can get this sample address verification URL from the XVerify back end system in the Reverse phone settings page.

Let's go through of each request and response fields one by one through the following table:

Request for Phone confirm verification service
--

Parameter in the Query String	Description
phone (required)	This request parameter contains the phone number for which we are interested to see the information
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
secret_key(optional)	This optional parameter if provided is used to automatically add a domain if not exist already.
Response for Phone confirm verification service	
name	This response field represents text of the listing name, both first and last, or the business name
firstname	This response field represents the first name of the residential listing (or the name of both people at the address, i.e. ("John & Mary")). This field is not populated for business listings.
lastname	This response field represents the last name of the residential listing. This field is not populated for business listings
fulladdress	This response field represents the full address of the person
houzenumber	This response field represents the text of the house number
housefraction	This response field represents the fractional portion of the house number. I.E. "1/2", or "B".
predirectional	This response field represents the directional that is placed in front of the street name; "SE", or "NW".
street	This response field represents the name of the street that the address is on
streettype	This response field represents the type of street that it is, i.e. Ave or Blvd
postdirectional	This response field represents directional after the name of the street; "NW" or "SE"
suiteno	This response field represents the suite, PO Box number or apartment number of the address
latitude	This response field represents the latitude, in degrees (i.e. 45.12345).
longitude	This response field represents the longitude, in degrees (i.e. -120.56789).
city	This response field represents the city of the listing.
state	This response field represents the state of the listing
zip5	This response field represents the first five digits of zip code
zip4	This response field represents the first four digits of zip code
npa	This response field represents the area code of listing phone number.

nxx	This response field represents the exchange of the listing phone number.
postfix	This response field represents the suffix of the listing phone number.
listingtype	This response field represents business, residential or government with “B”, “R”, or “G” respectively. <i>Note: listings will often be both business and residential, typically professionals working from home.</i>
flags	This response field represents different flags associated with the listing, such as non-pub or non-listed <ul style="list-style-type: none"> 1. NP – non-published 2. NL – non-listed 3. LP – listed and published record 4. OA – omit address 5. PA – populated address (mutually exclusive with OA) 6. FX – fax line 7. NF – non-fax line (mutually exclusive with FX) 8. CH – children’s line (only populated for residential listings) 9. NC – non-children’s line (only populated for residential listings, mutually exclusive with CH) 11. CS – Cellular Services, indicates a wireless number 12. PS - Paging Services, indicates a pager number
error	This field indicates error. If it’s non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means the system is under maintainance and will be available soon.
status	This response field has the following two possible values: <ul style="list-style-type: none"> 1. valid 2. unknown
message	This response field contains the actual message in human readable form.

Lead score Request/Response:

Here is the sample Lead Score service URL:

```
http://www.xverify.com/services/ scoring/verify/?phone=<phone_number>&
firstname=<first_name>& lastname=<last_name>&ip=<ip>& address=<address>&
city=<city>& state=<state>& zip=<zip>& email=<email>&
start_time=<start_time_in_milli_seconds>&ype=<xml/json>&&apikey=<api_key>&domain=<
all owed_domain>
```

We can get this sample lead score service URL from the XVerify back end system in the lead score settings page.

Let’s go through of each request and response fields one by one through the following table:

Request for Lead score service	
Parameter in the Query String	Description
phone (required)	This request parameter contains the phone number for which we are interested to see the information
firstname (required)	This request parameter contains the first name
lastname(required)	This request parameter contains the last name
address(required)	This request parameter contains the address
city (required)	This request parameter contains the city
state (required)	This request parameter contains the state
zip (required)	This request parameter contains the 5 digit zip
email (required)	This request parameter contains the email address
ip (required)	This parameter takes the IP address
start_time(required)	This is the time you will get automatically from the submitted form. You are just supposed to forward the same parameter to our system (i.e. Xverify)
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
secret_key (optional)	This optional parameter if provided is used to -automatically add a domain if not exist already.
Response for Lead score service	
Response field	Description
value	This response field represents the phone number provided by the user.
name	This response field indicates whether the name (i.e. Full Name) is matched or not. It's possible values are yes/no
	This response field indicates whether the last name is matched or not. It's possible values are yes/no
streetaddress	This response field indicates whether the address is matched or not. It's possible values are yes/no
validphone	If we found the provided phone as valid, we set this tag as "yes", otherwise "no"
phonetype	This response field contains the type of phone. It's possible values are "landline", "cell" and "voip".
validaddress	If we found the provided address as valid, we set this tag as "yes", otherwise "no"
city	This response field indicates whether the city is matched or not. It's possible values are yes/no

state	This response field indicates whether the state is matched or not. It's possible values are yes/no
zip	This response field indicates whether the 5 digit zip is matched or not. It's possible values are yes/no
iploc	This response field indicates whether the city we got in the IP lookup is matched with the user provided city or not. It's possible values are yes/no
validemail	This response field indicates whether the email actually exists or not. It's possible values are yes/no
score	This response field indicates the score in percentage.
error	This field indicates error. If it's non-zero, that means error occurred. You can lookup <i>message</i> response field to know the exact error. Please note that negative numbers are reserved for system messages. For example: If we have -1 value, that means the system is under maintainance and will be available soon.
status	This response field always has a "valid" value
message	This response field contains the actual message in human readable form.

All in One API Request/Response:

Here is the sample All in One API service URL:

```
http://www.xverify.com/services/allservices/verify/?services[email]=<email>&services[phone]=<phone_number>&services[address][street]=<street>&services[address][zip]=<zip>&services[address][city]=<city>&services[address][state]=<state>&domain=<allowed_domain>&type=<xml/json>&apikey=<api_key>
```

Let's go through of each request and response fields one by one through the following table:

Request for All in One API	
Parameter in the Query	Description
String	
services[email] (required)	This request parameter contains the email address to verify for email verification services. Please note that if this information is not provided or empty information is provided, our system will not use email verification service and ignore it.
services[phone] (required)	This request parameter contains the phone number to verify for phone verification services. Please note that if this information is not provided or empty information is provided, our system will not use phone verification service and ignore it.
services[address][street] (required)	This request parameter contains the street address. This is one of the required parameters in the address verification service. If

	all of the required information is not provided or empty information is provided, then, our system will not use address verification service and ignore it. However, if any one of the required information is missing, system will generate a proper response to let the user know about the missing parameter.
services[address][zip] <i>(required)</i>	This request parameter contains the zip code. This is one of the required parameters in the address verification service. If all of the required information is not provided or empty information is provided, then, our system will not use address verification service and ignore it. However, if any one of the required information is missing, system will generate a proper response to let the user know about the missing parameter.
services[address][city] <i>(optional)</i>	This request parameter contains the city. This is one of the optional parameters in the address verification service. If the optional parameters are missing, system will use the required information to verify the address.
services[address][state] <i>(optional)</i>	This request parameter contains the state. This is one of the optional parameters in the address verification service. If the optional parameters are missing, system will use the required information to verify the address.
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized unless we have test transactions count under per day limit
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
Response for All in One API	
Response field	Description
xvresponse	This is the root tag which contains all of the tags for the corresponding services.
email	This response field is a wrapper of email verification service response fields described in the above section named as " Email verification Request/Response "
phone	This response field is a wrapper of phone verification service response fields described in the above section named as " Phone verification Request/Response "
address	This response field is a wrapper of address verification service response fields described in the above section named as " Address verification Request/Response "

Postback URL Request/Response:

Here is the sample Postback URL:

```
http://www.xverify.com/uploads/post.php?  
first_name=[firstname]&last_name=[lastname]&age=[age]&gender=[gender]
```

Here, all of the query string parameters are defined by the user. User can define any number of parameters based on his business needs. To understand more, let's take the above example and discuss it further.

In the above example, we have defined the following query string parameters:

1. first_name
2. last_name
3. age
4. gender

and their values are enclosed in square brackets, also known as place holders, to give you the impression that these will be replaced by their actual values provided by the user. For such information, in the respective Xverify service settings page, user is just required to provide the parameter name and Xverify will handle the rest on the fly.

So, in one statement, all of the information which is not a part of Xverify service verification result, user is responsible to provide its parameter and their respective values (i.e. Through HTML or API Call). However, all of the information which is a part of Xverify service verification result, if the user wants his third party to use that information, he should define those parameters in the Xverify service settings page and Xverify will do the rest stuff automatically. That's all about its concept. Now, let's see how we can use this in both sides (i.e. HTML form and API call).

Let's assume that we have defined the following parameters in the email settings page:

1. first_name=[first_name_place_holder]
2. last_name=[last_name_place_holder]
3. age=[age_place_holder]
4. gender=[gender_place_holder]

Based on these parameters, our API call would look like something below:

```
http://www.xverify.com/services/emails/verify/?email=test@xverify.com&postback[first_name  
_place_holder]=Asim&postback[last_name_place_holder]=Arif&postback[age_place_holder]=3  
0&postback[gender_place_holder]=male
```

Now, let's take an example of HTML form. Based on our defined parameters in the email settings page, our DOM elements should look like the following:

1. <input type="text" class="xverify_postback" name="first_name_place_holder">
2. <input type="text" class="xverify_postback" name="last_name_place_holder">
3. <input type="text" class="xverify_postback" name="age_place_holder">
4. <input type="text" class="xverify_postback" name="gender_place_holder">

Furthermore, it's required to mention the class attribute named as "xverify_postback".

Affiliate Stats API Request/Response:

Here is the sample affiliate stat URL:

```
http://www.xverify.com/services/affiliate_stat/getAffiliateStats/? start_date=<start_date>& end_date=<end_date>&domain=<allowed_domain>&type=<xml/json>&apikey=<api_key>
```

Let's go through of each request and response fields one by one through the following table:

Request for Affiliate Stats		
Parameter in the Query String	Description	
start_date (required)	This request parameter contains the start date to get the affiliate/sub affiliate stats. Please note date format should be 'mm/dd/yyyy'	
end_date (required)	This request parameter contains the end date to get the affiliate/sub affiliate stats. Please note date format should be 'mm/dd/yyyy'	
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.	
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized	
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"	
Response for Affiliate Stats		
Response field	Description	
status	This response field contains either "valid" or "invalid"	
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.	
	Results are produced in XML with the primary object type being the "affiliates." Each affiliate will include all of the available fields related to the request.	
affiliates		This response field represents the relevant affiliate details on the criteria. This response field contains sub children 'affiliate'. Each affiliate include all the below sub children
	name	This child response field represents the complete name of the affiliate.
	status	This child response field represents affiliate status if affiliate blocked it contain 'block' otherwise empty.
	services	This child response field contain the services stats detail
	email	If you have subscribe email service you get this response field whose

			contains	
			valid_count	This response field get the valid email count for affiliate
			invalid_count	This response field get the invalid email count for affiliate
			disposable_count	This response field get the disposable email count for affiliate
			high_risk_count	This response field get the high risk email count for affiliate
			complainer_count	This response field get the complainer email count for affiliate
		phone	If you have subscribe phone service you get this response field whose contains	
			valid_count	This response field get the valid phone count for affiliate
			invalid_count	This response field get the invalid phone count for affiliate
		address	If you have subscribe address service you get this response field whose contains	
			valid_count	This response field get the valid address count for affiliate
			invalid_count	This response field get the invalid address count for affiliate

Sub Affiliate Stats API Request/Response:

Here is the sample sub affiliate stats URL:

```
http://www.xverify.com/ services/affiliate_stat/getSubAffiliateStats? start_date=<start_date>&
end_dat=< end_date>&domain=<allowed_domain>&type=<x ml/json>&apikey=<api_key>&
affiliate_id=< affiliate_id>
```

Let's go through of each request and response fields one by one through the following table:

Request for Sub Affiliate Stats	
Parameter in the Query String	Description
start_date (required)	This request parameter contains the start date to get the sub affiliate stats. Please note date format should be 'mm/dd/yyyy'
end_date (required)	This request parameter contains the end date to get the sub affiliate stats. Please note date format should be 'mm/dd/yyyy'
Affiliate_id (required)	This request parameter contains the affiliate id to get all of its sub affiliates
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be

	entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized	
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"	
Response for Sub Affiliate Stats		
Response field		Description
status		This response field contains either "valid" or "invalid"
message		This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
		Results are produced in XML/JSON tags with the primary object type being the "affiliates." Each affiliate will include all of the available fields related to the request.
sub_affiliates	This response field represents the relevant sub affiliate details on the criteria. This response field contains sub children 'sub_affiliate'. Each sub affiliate include all the below sub children	
	name	This child response field represents the complete name of the sub affiliate.
	status	This child response field represents sub affiliate status. If sub affiliate is blocked, then this response field contains 'block', otherwise empty.
	services	This child response field contains the subscribed services tags (i.e. "email" in case you are subscribed with email service)
	email	We will get the following child response tags only if the user is subscribed with email service.
	valid_count	This response filed get the valid email count for the particular sub affiliate
	invalid_count	This response filed get the invalid email count for the particular sub affiliate
	disposable_count	This response filed get the disposable email count for the particular sub affiliate
	high_risk_count	This response filed get the high risk email count for the particular sub affiliate
	complainer_count	This response filed get the complainer email count for the particular sub affiliate
	phone	We will get the following child response tags only if the user is subscribed with phone service.
	valid_count	This response filed gets the valid phone count for the particular sub affiliate
	invalid_count	This response filed get the invalid phone count for the particular sub affiliate
	address	We will get the following child response tags only if the user is subscribed with address service.
	valid_count	This response filed get the valid address count for the particular sub affiliate
	invalid_count	This response filed get the invalid address count for the particular sub affiliate

Add Domain API Request/Response:

Here is the sample Add Domain URL:

```
http://www.xverify.com/ services/add_domain/? domain=<new_domain_name>&type=<xml/json>&apikey=<api_key>
```

Let's go through of each request and response fields one by one through the following table:

Request for Add Domain	
Parameter in the Query String	Description
apikey (required)	This parameter takes the API key. API key is a value which we get in XVerify system once we are successfully registered with it.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized
type (required)	This parameter can take either "xml" or "json". If we want to get the response in XML, then we need to pass "xml", otherwise, we need to pass "json"
Response for Add Domain	
Response field	Description
error	This response field contains either "0" or "1", 1 mean getting error to add domain in user account
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.

Add Postback Job Request/Response:

This API is very handy because through this API, you can submit a batch of email addresses through API that would be processed in the background. Maximum number of email addresses per call is 1000, when you send more than 1000 in request we reject the request now. Due to it's batch nature, a job id is returned right away that can be further looked up in the separate API explained later. This API accepts JSON data in HTTP POST method

Here is the sample Add Postback Job URL:

```
http://www.xverify.com/v1/emails/postback/
```

And here is the sample JSON data:

```
{"emails":[{"email":"abc@gmail.com","data":{"id":123,"name":"John Doe"}},{"email":"abc@hotmail.com"}, {"email":"abc@yahoo.com"}, {"email":"xyz@gmail.com"}, {"email":"robppc@yahoo.com"}], "type":"json", "apikey":"<api_key>", "domain":"<domain_name>"}
```

Let's go through of each request and response fields one by one through the following table:

Request for Add Postback Job	
Parameter in the Query String	Description
type (optional)	This optional parameter must contain json as value.
domain (required)	This parameter takes the allowed domain for which this query will be entertained. Please keep in mind that you need to add one or more domains through the XVerify system which you will provide here. We won't entertain a query for which this value is unrecognized.
auto_correct(optional)	This optional parameter tells we need to auto correct the email. its possible value is '0' or '1'. When provide '1' it will be correct the email, if the is parameter not provided then use the domain setting.
callback_url (optional)	This optional parameter tells we send the call back request when verify all the emails. When this parameter not provided then use callback url value from the domain setting page.
apikey (required)	This required parameter contains the xverify provided api key.
emails (required)	This required parameter contains a list of emails where each email belongs to email tag. This tag may also contain optional data tag which is what returned as it is in each email once it's processed
email (required)	This required parameter contains the email address to be processed
data (optional)	This optional parameter contains the extra data that can be supplied by the caller along each email tag and it will be returned as it is with each email once processed.
auto_correct(optional)	This optional parameter tells we need to auto correct the email which can be supplied by the caller along each email tag and its possible value is '0' or '1'. When provide '1' it will be correct the email, if the is parameter not provided then check the root level auto correct value if not provided then use the domain setting.
Response for Add Postback Job	
Response field	Description
error	This response field contains either "0" or "1", "1" is returned in case of any error. If all goes well, we return with "0".
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
responsecode	This contains 1 on success or the respective error code
status	This contains either success on success or the respective failure status on failure case.
postback_job_id	This contains the submitted job id that can be further looked up for it's progress in the separate API explain below

Get Postback Job Status Request/Response:

This API is used to lookup the status of the submitted postback job. It accepts data in HTTP GET/POST method. In POST, data could also be submitted as JSON string

Here is the sample Get Postback Status Job URL:

```
http://www.xverify.com/v1/emails/getpostbackstatus?apikey=<api_key>&postback_ids[]={<postback_id_1>&postback_ids[]={<postback_id_2>&postback_ids[]={<postback_id_3>
```

Let's go through of each request and response fields one by one through the following table:

Request for Get Postback Status Job	
Parameter in the Query String	Description
type (optional)	This optional parameter can contain either xml or json.
apikey (required)	This required parameter contains the xverify provided api key.
postback_ids [](required)	This required parameter contains a list of postback job ids for which the lookup is desired
Response for Get Postback Status Job	
Response field	Description
error	This response field contains either "0" or "1", "1" is returned in case of any error. If all goes well, we return with "0".
message	This response field contains the actual message in human readable form. In case of error, the same message is prompted to the user.
responsecode	This contains 1 on success or the respective error code
postback	This parent tag contains request tags as child where each child represents information about each postback job
request	This contains information about a particular postback job. This tag is repeated for each postback job
id	This is a child tag of request tag which represents postback job id
status	This is a child tag of request tag which represents current status of the postback job. It's potential values are pending, in progress and completed
total	This is a child tag of request tag which represents total submitted email addresses for the particular postback job.
type	This is a child tag of request tag which currently returns email only which means it's a request of email job.
valid_count	This is a child tag of request tag which represents total valid email count for the particular postback job.
invalid_count	This is a child tag of request tag which represents total invalid email count for the particular postback job.
unknown_count	This is a child tag of request tag which represents total unknown email count for the particular postback job.
processed_rows	This is a child tag of request tag which represents total processed count for the particular postback job.

Get Postback Job Detail Request/Response:

This API is used to lookup the detailed information of the submitted postback job. It accepts data in HTTP GET/POST method. In POST, data could also be submitted as JSON string

Here is the sample Get Postback Status Job URL:

```
http://www.xverify.com/v1/emails/getPostbackDetails?apikey=<api_key>&postback_id=<postback_job_id>
```

Let's go through of each request and response fields one by one through the following table:

Request for Get Postback Detail Job	
Parameter in the Query String	Description
type (optional)	This optional parameter can contain either xml or json.
apikey (required)	This required parameter contains the xverify provided api key.
Postback_job_id (required)	This required parameter contains a postbackjob id for which the detail is required.
Response for Get Postback Detail Job	
Response field	Description
status	This response field represents the current status of the postback job id. Potential values are pending, in progress and completed.
emails	This response field contains multiple email tags where each tag corresponds to particular email to be processed
email	This response tag is a child tag of the emails tag and contains information about a particular email address. All of it's child tags are exactly the same like in the email verify API "verify"