TowerData Real-Time Validation & Enhancement API

Table of Contents

1 Introduction

The TowerData Real-Time Validation and Enhancement Application Programming Interface (API) is a hosted service that improves the quality of the customer contact information you collect, finds additional points of contact, and provides information about who and where your customers are. It can be integrated into a website where contact information is collected or into a backend server that is processing customer data.

1.1 Available Services

The API offers the following services depending on the point of contact.

Contact point	Services Available
Email Addresses	• Validation • Correction • Append name and address • Append demographics
Postal Addresses	• Validation • Correction • Append demographics
Phone Numbers	• Validation • Append location • Append name and address
IP Address	• Append location and ISP

1.2 Using the API

Using the TowerData Validation and Enhancement API is as simple as constructing a URL. The information you want to process is contained within the URL and is submitted to TowerData's servers. TowerData will process the data and return the results usually within one second.

Here's how it works:

1. A website or program makes a request to the API using the standard HTTP protocol.
2. Each request contains one or more pieces of data regarding a single individual and specifies how the data should be processed.
3. TowerData's servers perform the requested services on the data received.
4. The results of the services are gathered and returned in a single response in JSON format to the calling program.
5. The website or program that made the request parses the response and uses the results with appropriate business logic for your company.

Accessing the API can be done in clear text or encrypted using HTTPS.

1.3 Creating an Account

Contact your TowerData sales representative to create an account and get access to the API.

2 Calling the API

2.1 API Resource

All calls to the API shall be made to the following URL location:
http://api10.towerdata.com/person

2.2 API Parameters

The following parameters are accepted by the API.

Parameter	Description
license	The authorization key for accessing the services. Contact your sales representative for yours. Access to the API can also be controlled by IP address or domain.
email	Email address
phone	Phone number
fname	First name
lname	Last name
address1	First line of postal address
address2	Second line of postal address
city	City of postal address
state	Two letter state abbreviation of US or Canadian postal address
zip	Zip code of postal address in 5 digit or 9 digit format
urbanization	Urbanization name for Puerto Rican postal addresses only
ip	IP address. If the value of the ip parameter is “detect”, the service will use the IP address of the machine connecting to it.
demos	Requests demographic based on the submitted contact information. You will receive a separate API key for demographics that must be supplied with this parameter.
validate	Specifies which pieces of information provided should be validated. Valid values are "email," "email-domain-only", "phone," “phone-full”, "postal," and “none." A value of "email-domain-only" means that the service will only check the syntax and domain of an email address. A value of "email" means that that syntax, domain, and the mailbox will be checked. A value of “phone” means that the service fill validate the first 7

	digits of a phone number, whereas with a value of “phone-full” the service will compare a 10 digit phone number with the American 411 directory. Values can be combined with a plus sign, e.g. "email+phone". If this parameter is omitted, all information in the request will be validated. Nothing will be validated if the validate parameter has a value of “none”.
find	Specifies the type of information to find based on the input data. Valid values are "postal", to find name and address based on email address or phone number (email is tried first); “postal-email”, to find name and address by email address only; or “postal-phone”, to find name and address by phone number only.
correct	Specifies which pieces of information provided should be corrected, if they are invalid. The only valid value is "email". Postal information will always be corrected if it is validated.
timeout	The maximum number of seconds the request should take. Default value is 5 seconds, and maximum allowed value is 500 seconds. Optional.
format	The output format of the results: json or html. Default is json. HTML can be selected for displaying the results in a web browser.
log	If set to true, the inputs and the outputs of the service will be stored on TowerData servers for debugging purposes.

Please note:

- ***All values supplied in the request must be URL encoded***
- Only include parameters that are needed for the information you want to process
- The casing of parameter values is ignored

2.3 Constructing an API Request

Calling the API involves constructing a URL with the appropriate resource and parameters and invoking the URL with HTTP GET.

All API requests start with:

`http://api10.towerdata.com/person?`

Note the question mark (`?`) on the end.

Append the parameters you want to use to the resource in `name=value` pairs. Each value must be URL encoded so that any blank spaces or special characters are turned into symbols

that will work within a URL. Values can be in upper, lower, or mixed case. Multiple parameters must be separated by an ampersand (`&`).

You only need to include the parameters that are needed for the information you want to process.

Here is an example API query to validate the email example@domain.com:

```
http://api10.towerdata.com/person?email=example%40domain.com&license=12345678
```

There are only two parameters in the above request, email and license, and the "at" sign (@) in the email has been encoded with the `%40` symbol.

To process this information:

Email	example@domain.com
First name	John
Last name	Doe
Address	100 Main St
City	New York
State	NY
Zip	10001
Phone	(212) 5551212
IP Address	74.125.113.103

and request demographics, the query would look like:

```
http://api10.towerdata.com/person?email=example%40domain.com&license=12345678&phone
=(212)%205551212&fname=john&lname=doe&address1=100%20main%20st&city=new%20york&stat
e=ny&zip=10001&ip=74.125.113.103&demos=YOURDEMOKEY
```

All spaces in the input data have been URL encoded to the symbol `%20`.

2.4 Testing an API Request

You can type a URL to access the API directly into a web browser to see the results. We soon will be providing an option to print the output in a more readable format.

From a UNIX command line, you can use curl to try a URL. Put the URL within single quotes. For example:

```
$ curl 'http://api10.towerdata.com/person?email=example%40domain.com&license=123456'
```

3 API Output

3.1 Response Output

The response from the API will contain a section, which is represented as a JSON object, for each type of data returned. The sections that may be returned are:

- "email" – Email validation
- "phone" – Phone validation
- "postal" – Postal validation
- "found_postal" – Postal address found from email (reverse email append)
- "demographics" – Demographics
- "ip" - IP address geolocation

In JSON, objects and the values within them are unordered. Thus, programs calling the API should never make assumptions about the sequencing of the data returned by it.

Each section will always contain the following three elements

- "ok" – Indicates the overall status of the section using JSON native types with three possible values
  - true – Data is valid or data requested was found
  - false – Data is invalid or requested data was not found
  - null – A timeout or error occurred while processing the request
- "status_code" – A numeric value that indicates the status of the section in greater detail. The values used can vary from section to section.
- "status_desc" – A readable text description of the status.

In addition, all of the JSON objects representing each data section will be enclosed in a JSON object. This outer object will contain "status_code" and "status_desc" elements as well, which indicate the overall outcome of the request.

For example, here is a condensed version of a response with an email and phone sections.

```
{
  "email": {
    "ok": false,
    "status_code": 110,
    "status_desc": "Invalid character in address",
    "address": "john;doe@example.com",
    },
  "phone": {
    "ok": true,
    "status_code": 10,
    "status_desc": "Valid area code and exchange",
    "number": "3038322478",
  },
  "status_code" : 10,
  "status_desc" : "Successful Request"
}
```

3.2 Parsing API Responses

Client programs calling the API

- Should not depend on the ordering of any elements with a section.
- Should ignore response elements they do not recognize.
- Should not rely on all elements existing in a response section if the status_code for that section indicates an error or no data found.

3.3 Email Validation and Correction

The email component of the API performs the following escalating series of checks to validate an email address:

- Detects format and syntax errors
- Ensures the address use a valid top-level domain (e.g. .com, .net, .biz, etc.)
- Detects improper email address formats for common domains such as Hotmail and AOL
- Verifies whether the domain of the email addresses exists
- Confirms that the domain can receive email
- Determines whether the user exists at the domains

Information about an email address is returned in the `email` output section. The "email" parameter must be included in the request to receive email output, and the "correct" parameter must be included to receive suggested corrections for invalid emails.

Checking that an address exists at a domain can potentially take a couple of seconds because it requires the service to connect to the domain and test the mailbox. Checking only the syntax and domain always takes less than a second and can be done by providing a value of "email-domain-only" in the "validate" parameter of the request.

Response elements that may be included in the email section:

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.3.
status_desc	Description of status. See 4.3.
address	The email address
username	The portion of the email address to the left of the @
domain	The portion of the email address to the right of the @
validation_level	The actual level of validation performed on the email address. 0 = Timeout occurred 2 = Syntax and domain checked against database 3 = Syntax and domain existence confirmed in real-time 4 = Syntax and domain receives email confirmed in real-time 5 = Syntax, domain, and mailbox checked

domain_type	A classification of the type of domain used in the email address. “disposable” = The domain belongs to a disposable or temporary email service. You may wish to block emails of this type. “wireless” = The email is for a wireless service and the FCC prohibits sending unsolicited emails to these domains. “null” = The type is unknown.
role	A true/false flag indicating whether the email address is a role-based account, such as sales@ or info@.
corrections	For an incorrect email address, an array with possible correct email addresses, if a correction could be made. Only present if the `correct=email` parameter is in the request.

Example response for a valid email:

```
{
  "email":{
    "ok": true,
    "status_code": 50,
    "status_desc": "Domain does not reject the mailbox"
    "validation_level": 5,
    "address": "sales@attmobility.net",
    "username": "sales",
    "domain": "example.com",
    "domain_type": "wireless",
    "role": true
  }
}
```

Example response for an invalid email with suggested corrections for the email obtained by using the `correct=email` parameter:

```
{
  "email": {
    "ok": false,
    "status_code": 110,
    "status_desc": "Invalid character in address",
    "validation_level": 2,
    "address": "john;doe@example.com",
    "username": null,
    "domain": null,
    "domain_type": null,
    "role": false
    "corrections": ["john.doe@example.com",
                    "johndoe@example.com"]
    }
  }
}
```

Example response if the service was not able to validate the email within the timeout period:

```
{
```

```
  "email": {
    "ok": null,
    "status_code": 5,
    "status_desc": "Validation Timeout"
    "validation_level": 0,
    "address": null,
    "username": null,
    "domain": null,
  }
}
```


3.4 Reverse Email Append

The API can provide the person's name and postal address that corresponds to an email address. The "email" and either the "find=postal" or the “find=postal-email” parameters must be included in the request to receive this section in the response.

Response elements that may be included in the found_postal section:

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.6.
status_desc	Description of status. See 4.6.
fname	First name of the individual
lname	Last name of the individual
address1	First address line
address2	Second address line
city	City
state	U.S. state abbreviation
zip	U.S. zip code (5 digits)
plus4	U.S. 4 digit zip addon
source	Indicates how the postal information was found. Value = “email”

Example found_postal output:


```
{
  "found_postal": {
    "ok": true,
    "status_code": 10,
    "status_desc": "Data found",
    "fname": "JANE",
    "lname": "SMITH",
    "address1": "100 MAIN ST APT 3",
    "address2": null,
    "city": "SPRINGFIELD",
    "state": "MA",
    "zip": "11111",
    "plus4": "1234",
    "source": "email"
  }
```

```
}
```

Example, if no information is found.

```
{
  "found_postal": {
    "ok": false,
    "status_code": 15,
    "status_desc": "No data found"
  }
}
```

3.5 Demographic Append

The API can provide dozens of different demographic characteristics about an individual or household based on their name and postal address *or* their email address.

Demographic information is returned in the `demographics` output section. The "demos" parameter must be included in the request to receive this section in the response along with the appropriate input information (email address and/or name and postal address). Your sales representative will provide you with a separate API key that you must provide as the value for the parameter, e.g., "demos= 098f6bcd4621d373cade4e832627b4f6". For the greatest match rate, provide email, name, and address together. If you do not have address, still provide first and last name when it is available.

Below is the list of elements available in the demographics section. Contact your sales representative to configure which elements you receive.

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.7.
status_desc	Description of status. See 4.7.
age	Age of the individual in ranges. 18-20 21-24 25-34 35-44 45-54 55-64 65+
gender	Gender of the input individual: “Male” or “Female”
zip	“90210”, …
And over 30 more	For a full list of demographic fields and values, refer to our Data Dictionary.

Demographic elements will only appear in the output if there is a value available for that element.

Example demographics output:

```
{
   "demographics" : {
      "household_income" : "25k-50k",
      "zip" : "32707",
      "gender" : "Male",
      "status_desc" : "Data found",
      "ok" : true,
      "age" : "55-64",
      "home_owner_status" : "Own",
      "length_of_residence" : "20+ years",
      "education" : "Completed College",
      "interests" : {
        "Travel" : true,
        "Automotive" : true,
        "Charitable Donors" : true,
        "Sports" : true,
        "Technology" : true
      }
   }
}
```

Example demographics output if no information is found:

```
{
  "demographics": {
    "ok": false,
    "status_code": 15,
    "status_desc": "No data found",
  }
}
```

3.6 Phone Validation

The telephone component of the API can validate a phone number for the United States, Canada and 15 other countries. The Telephone Validation service will:

- Check if the area code is valid and whether the exchange, which is the first three digits of a seven digit number, is valid for that area code.
- Provide detailed regional information for the phone number including: time zone, daylight savings time, country, state, county, city, longitude, and latitude.
- Supply a new area code for a number which has or is being updated.
- Determine if the phone number is listed in the 411 directory of the United States, if the “phone-full” parameter is specified.

Information returned about a telephone number is returned in the `phone` output section. The "phone" parameter must be included in the request to receive this section.

Response elements that may be included in the phone section:

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.4.
status_desc	Description of status. See 4.4.
number	The standardized phone number
extension	The standardized extension, if applicable
new_npa	The new area code for the phone number, if applicable. If the area code is changing, the existing area code may still be accepted as valid for a period of time.
timezone	The time zone the telephone number is in as an offset from GMT
observes_dst	Flag indicating whether the time zone observes daylight savings time. 1 = Observes DST (Daylight Saving Time) during the normal DST observance period, 0 = Does not observe DST.
country	Two character International Standard ISO 3166-1 Country Code for the phone number: US - United States CA - Canada BS - Bahamas BB - Barbados AI - Anguilla AG - Antigua and Barbuda VG - Virgin Islands, British KY - Cayman Islands BM - Bermuda GD - Grenada TC - Turks and Caicos Islands MS - Montserrat LC - Saint Lucia DM - Dominica VC - Saint Vincent and the Grenadines DO - Dominican Republic TT - Trinidad and Tobago KN - Saint Kitts and Nevis JM - Jamaica
state	Two letter USPS state abbreviation (US, Canada and US territories)
county	County name
city	City name or general location
latitude	Latitude in decimal degrees of the center of the area covered by the phone number

longitude	Longitude in decimal degrees of the center of the area covered by the phone number
line_type	Indicates the telephone service type the phone number was originally designated for. Due to number porting, individual phone numbers may have switched type. Valid values are L = Land line, non-wireless service C = Wireless type services P = Paging and other messaging services M = Mixed wireless and land line T = Toll free
carrier	The name of the telecommunications company the phone number was originally assigned to. Due to number porting, individual phone numbers may have switched carrier.
messages	An array of JSON objects containing additional comments on the phone number, if applicable. Each object will contain a `"code"` and a `"desc"` element. See codes 140 and higher in Section 4.4.

Example output for a valid phone number:

```
{
  "phone": {
    "ok": true,
    "status_code": 10,
    "status_desc": "Valid area code and exchange",
    "number": "3038322478",
    "extension": null,
    "state": "CO",
    "city": "DENVER",
    "new_npa": null,
    "country": "US",
    "county": "DENVER",
    "latitude": "39.73",
    "longitude": "-104.97",
    "timezone": "-7",
    "observes_dst": "1",
    "line_type": "L",
    "carrier": "QWEST",
    "messages": [{"code":140, "desc": "Extension too long"},
                 {"code":150, "desc": "Toll Free number"}
                ]
  }
}
```

Example output for an invalid telephone number:

```
{
  "phone": {
    "ok": false,
    "status_code": 120,
    "status_desc": "Too few digits",
```

```
    "detail_level": 3,
    "number": null,
    "extension": null,
    "state": null,
    "new_npa": null,
    "country": null,
    "county": null,
    "latitude": null,
    "longitude": null,
    "timezone": null,
    "observes_dst": null,
    "line_type": null,
    "carrier": null,
    "messages": []
  }
}
```


3.7 Postal Validation and Standardization

The API can validate and correct U.S. and Canadian postal addresses using a DPV® and SERP Certified process. In addition to standardizing the address, the service will provide all the information you need, such as carrier route and delivery point code, to obtain postal discounts.

Information about a postal address is returned in the `postal` section. The "address1" parameter and either the "city" or the "zip" parameter must be included in the request to receive this section in the response.

Response elements that may be included in the postal section:

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.5.
status_desc	Description of status. See 4.5.
address1	First address line
address2	Second address line
city	City
state	U.S. state abbreviation or Canadian province code
zip	U.S. zip code (5 digits) or Canadian postal code
plus4	U.S. 4 digit zip addon
type	Record type: (F)irm, (G)eneral delivery, (H)ighrise, (P)O Box, (R)ural route/Highway contract, (S)treet
carrier_route	Carrier route
elot	Line of travel number. Four digit sequence number plus a directional indicator.
dpc_cd	2 digit delivery point code plus check digit

county	Name of the county
county_num	Last 3 characters of FIPS code
urbanization	Urbanization for Puerto Rican addresses
dpv	Delivery Point Confirmation (DPV) Y = Address DPV confirmed for both primary and (if present) secondary numbers D = Address DPV confirmed for the primary number only, and secondary number information was missing S = Address DPV confirmed for the primary number only, and secondary number information was present but unconfirmed N = Both primary and (if present) secondary number information failed to DPV confirm Blank = Address not presented to hash table.
dpv_notes	DPV footnotes. Up to 12 bytes. AA: Input Address Matched to the ZIP+4 file A1: Input Address Not Matched to the ZIP+4 file BB: Input Address Matched to DPV (all components) CC: Input Address Primary Number Matched to DPV but Secondary Number not Matched (present but invalid) N1: Input Address Primary Number Matched to DPV but Address Missing Secondary Number M1: Input Address Primary Number Missing M3: Input Address Primary Number Invalid P1: Input Address RR or HC Box number Missing P3: Input Address PO, RR, or HC Box number Invalid RR: Input Address Matched to CMRA and PMB designator present (PMB 123 or #123) R1: Input Address Matched to CMRA but PMB designator not present (PMB 123 or #123) F1: Input Address Matched to a Military Address G1: Input Address Matched to a General Delivery Address U1: Input Address Matched to a Unique ZIP Code

Example postal output:


```
{
  "postal": {
    "ok": true,
    "status_code": 10,
    "status_desc": "Deliverable address",
    "address1": "100 MAIN ST APT 3",
    "address2": null,
    "city": "SPRINGFIELD",
    "state": "MA",
    "zip": "11111",
    "plus4": "1234",
```

```
    "urbanization": null
    "type": "H",
    "carrier_route": "C007",
    "elot": "0127A",
    "dpc_cd": "992",
    "county": "SAN FRANCISCO",
    "county_num": "075",
    "dpv": "D",
    "dpv_notes": "AAN1"
  }
}
```


3.8 Reverse Phone Append

The API can provide the person's name and postal address that corresponds to a phone number. The "phone" and either the "find=postal" or the “find=postal-phone” parameters must be included in the request to receive this section in the response. Only United States phone numbers are supported.

Results will be returned in a section called found_postal_phone which may contain the following elements:

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.6.
status_desc	Description of status. See 4.6.
source	Indicates how the postal information was found. Value = “phone”
results	If data was found, a results array that contains one or more listings associated with the phone number will be present. Each listing in the array will contain the elements below.
fname	First name of the individual
lname	Last name of the individual
name	For residential listings, last name followed by first name. For business listings, company name.
address1	First address line
address2	Second address line
city	City
state	U.S. state abbreviation
zip	U.S. zip code (5 digits)
plus4	U.S. 4 digit zip addon

Example found_postal_phone output:


```
{
  "found_postal_phone": {
    "ok": true,
    "status_code": 10,
```

```
    "status_desc": "Data found",
    "source": "phone",
    "results":[
      {
            "fname": "JANE",
            "lname": "SMITH",
            "name": "SMITH JANE",
            "address1": "100 MAIN ST APT 3",
            "address2": null,
            "city": "SPRINGFIELD",
            "state": "MA",
            "zip": "11111",
            "plus4": "1234",
      },
            "fname": null,
            "lname": null,
            "name": "JANE'S STORE",
            "address1": "100 MAIN ST APT 3",
            "address2": null,
            "city": "SPRINGFIELD",
            "state": "MA",
            "zip": "11111",
            "plus4": "1234",
      }
    ]
  }
}
```


Example, if no information is found.


```
{
  "found_postal_phone": {
    "ok": false,
    "status_code": 15,
    "status_desc": "No data found",
    "source": "phone",
  }
}
```


3.9 IP Address Geolocation

The API can identify where one of your customers is physically located while accessing your website and what ISP they are using by examining their IP address.

Information about an IP address is returned in the `ip` output section. The "ip" parameter must be included in the request to receive this section in the response. If you're calling the API directly from the browser of the user using JavaScript, the API can determine the IP address of the user if you specify `ip=detect` in the request.

Response elements that may be included in the IP section:

Element	Description
ok	Overall status of section. See 3.1.
status_code	Detailed status code. See 4.8.
status_desc	Description of status. See 4.8.
address	IP address
country_code	Two character ISO 3166 country code
country_name	Name of country from ISO 3166
region	Region or state name
city	Name of city
ISP	Internet Service Provider or company name
domain	Domain name

Example IP output:

```
{
  "ip": {
    "ok": true,
    "status_code": 10,
    "status_desc": "IP Found",
    "address": "66.108.76.101",
    "country_code": "US",
    "country_name": "UNITED STATES",
    "region": "NY",
    "city": "NEW YORK",
    "isp": "TIME WARNER CABLE",
    "domain": "roadrunner.com"
  }
}
```

Example response for an invalid IP:

```
{
  "ip": {
    "ok": false,
    "status_code": 100,
    "status_desc": "Invalid format",
    "address": "66.108.76",
    "country_code": null,
    "country_name": null,
    "region": null,
    "city": null,
    "isp": null,
    "domain": null
  }
}
```

3.10 Multiple Outputs in a Response

If more than one service of the API is invoked, a section corresponding to each service as described above will be included in the output.

Example with email, phone, postal, demographic and IP address output.


```
{
  "email": {
    "ok": false,
    "status_code": 110,
    "status_desc": "Invalid character in address",
    "validation_level": 2,
    "address": "john;doe@example.com",
    "username": null,
    "domain": null,
    "domain_type": null,
    "role": false,
    "corrections": ["john.doe@example.com",
                    "johndoe@example.com"]
    },
  "phone": {
    "ok": true,
    "status_code": 10,
    "status_desc": "Valid area code and exchange",
    "detail_level": "3",
    "number": "3038322478",
    "extension": null,
    "state": "CO",
    "new_npa": null,
    "country": "US",
    "county": "DENVER",
    "latitude": "39.73",
    "longitude": "-104.97",
    "timezone": "-7",
    "observes_dst": 1,
    "line_type": "L",
    "carrier": "QWEST",
    "messages": [{"code":140, "msg": "Extension too long"},
                 {"code":150, "msg": "Toll Free number"}
                ]
  },
  "postal": {
    "ok": true,
    "address1": "100 MAIN ST APT 3",
    "address2": null,
    "city": "SPRINGFIELD",
    "state": "MA",
    "zip": "11111",
    "plus4": "1234",
    "type": "H",
    "carrier_route": "C007",
    "lot_num": "0127A",
    "check_digit": "992",
    "county": "SAN FRANCISCO",
    "county_fips": "075",
    "dpv": "D",
    "dpv_notes": "AAN1"
```

```
  },
  "demographics" : {
    "household_income": "25k-50k",
    "zip": "32707",
    "gender": "Male",
    "status_desc": "Data found",
    "ok": true,
    "age": "55-64",
    "home_owner_status": "Own",
    "length_of_residence": "20+ years",
    "education": "Completed College",
    "interests": {
      "Travel": true,
      "Automotive": true,
      "Charitable Donors": true,
      "Sports": true,
      "Technology": true
    }
  },
  "ip": {
    "ok": true,
    "address": "66.108.76.101",
    "country_code": "US",
    "country_name": "UNITED STATES",
    "region": "NY",
    "city": "NEW YORK",
    "isp": "TIME WARNER CABLE",
    "domain": "roadrunner.com"
  },
  "status_code": 10,
  "status_desc": "Success"
}
```


3.11 Failed request output

If the request failed for some reason, one of the HTTP status codes (See 4.1) will be returned, and the body of the response will contain an "overall response" code (See 4.2).

For example:


```
{
  "status_code": 100,
  "status_desc": "Invalid request"
}
```


4 Status Codes

The status codes that may be returned by the service along with their descriptions.

4.1 HTTP status codes

These are the HTTP codes the service will explicitly return, however, client programs should be prepared to handle any of the standard HTTP codes.

HTTP Code	Description
200 OK	Request processed successfully
400 Bad Request	Some part of the request is invalid
401 Unauthorized	License key was missing or invalid
403 Forbidden	License key has expired, has exceeded the number of licensed requests, or does not have access to the feature requested
500 Internal Server Error	An unexpected error occurred on the server

4.2 Overall response codes

These are the codes that are returned at the outermost layer of the service.

Code	Description
5	One or more services timed out
8	One or more services failed
10	Success
100	Invalid request
110	Invalid parameter
200	Not authorized for one or more services
999	Service unavailable

4.3 Email codes

Code	Description
5	Timeout
10	Syntax OK
20	Syntax OK and domain valid according to the domain database
30	Syntax OK and domain exists
40	Syntax OK, domain exists, and domain can receive email
45	Domain does not support validation (accepts all mailboxes)
50	Syntax OK, domain exists, and mailbox does not reject mail
100	General syntax error
110	Invalid character in address

115	Invalid domain syntax
120	Invalid username syntax
125	Invalid username syntax for that domain
130	Address is too long
135	Address has unbalanced parentheses, brackets, or quotes
140	Address does not have a username
145	Address does not have a domain
150	Address does not have an @ sign
155	Address has more than one @ sign
200	Invalid top-level-domain (TLD) in address
205	IP address not allowed as a domain
210	Comments are not allowed in email addresses
215	Unquoted spaces are not allowed in email addresses
310	Domain does not exist
315	Domain does not have a valid IP address
325	Domain cannot receive email
400	The mailbox is invalid or the username does not exist at the domain
410	Mailbox is full and can not receive email at this time
420	Mail is not accepted for this domain
500	Addresses with that username are not allowed
505	Addresses with that domain are not allowed
510	Address is a bot or other suppression
512	Address is a role-based account
520	Address is a known bouncer
525	Address is a spamtrap or known complainer
530	Address has opted out from commercial email
999	Internal error

4.4 Phone codes

Code	Description
5	Timeout
10	Valid area code and exchange
50	Number listed in 411
100	Invalid exchange for area code
110	Invalid area code and exchange
120	Invalid number – too few digits
130	Invalid number – too many digits
133	Error – Exchange and number not allowed

134	Error – Exchange not allowed
135	Error – Phone number not allowed
140	Warning – Extension is greater than 5 digits in length
150	Warning – Toll free number
160	Warning – 900 toll number
999	Internal error

4.5 Postal codes

Code	Description
5	Timeout
10	Deliverable address
100	Multiple matches
110	Invalid delivery point – 5 digit zip returned
120	Invalid address – Address standardized
130	Invalid address – Original address returned
120	Invalid address – Address standardized
150	Invalid address – Insufficient input data
999	Internal error

4.6 Found_Postal and Found_Postal_Phone codes

Code	Description
5	Timeout
10	Data found
15	No data found
999	Internal error

4.7 Demographics codes

Code	Description
5	Timeout
10	Data found
15	No data found
100	Insufficient parameters
110	Invalid demographics API key
999	Internal error

4.8 IP Address codes

Code	Description
5	Timeout
10	IP found
15	IP not found
100	Invalid IP address
999	Internal error