



SWIFT EMAIL VERIFIER API DOCUMENTATION

V4.0

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT US FOR A COPY.

IN NO EVENT SHALL SWIFT OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF SWIFT OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SWIFT EMAIL VERIFIER: API Documentation V4.0
Copyright© Swift Software & Services. All rights reserved.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Swift Software & Services and any other company.

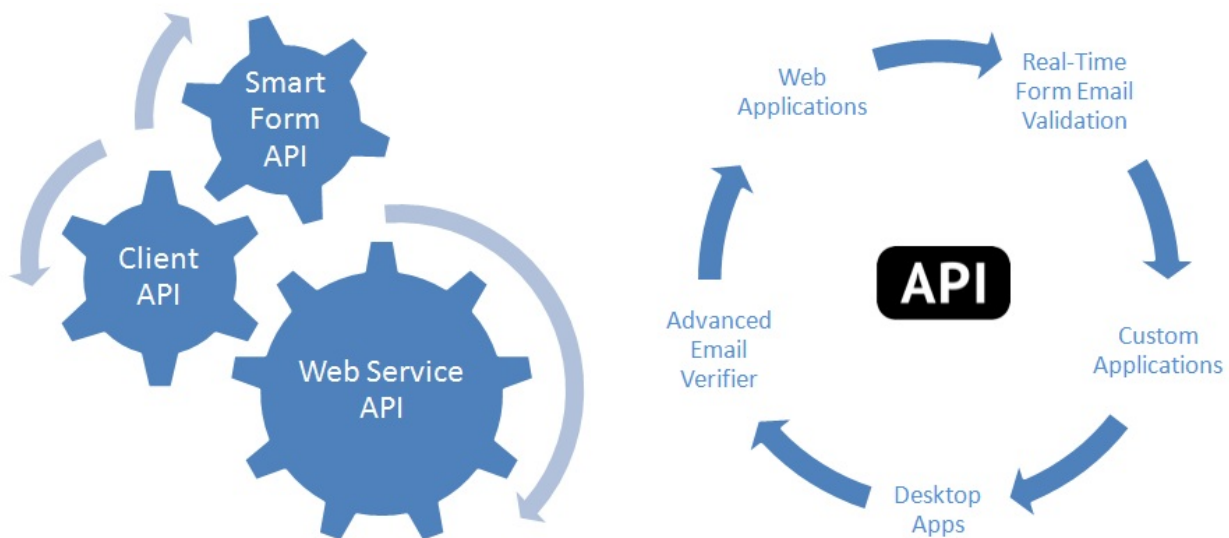
Contents

HOW IT WORKS.....	2
OVERVIEW OF EMAIL VALIDATION API.....	3
EMAIL VALIDATION STATUS AND STATUS CODES	3
RESOURCE URL.....	6
AUTHENTICATION	7
HTTP STATUS CODES	7
REPRESENTATIONS.....	7

HOW IT WORKS

Swift email validation API is a RESTful API that allows consumers and developers to verify or scrub email lists and process email bounces thereby enabling users to maintain a clean mailing list for your email marketing campaigns and reduce bounces to their mailbox. Our email validation service performs a complete and full verification by checking if an email address actually exist (actual mailbox existence) and can receive email (deliverable) at that specific moment in time.

Where we are unable to conclusively verify that an email address actually exists AND can receive email at that instant it was verified, we will respond back with an "Unknown" status. Unknown email address status does not attract any charges.



OVERVIEW OF EMAIL VALIDATION API

Our real-time email validation APIs allows you to check if an email address really exists and if it can receive messages. For every email address checked, a specific status is presented which tells you if the email address is valid or invalid or whether it is damaging or undesirable for your email marketing including over [20 status codes](#) for investigating the reason of a specific email validation failure.

What is Checked by Email Validation API (In progressive order):

- ✓ **Syntax, Typo and Fake Email Pattern Match:** This checks for syntax and for typo errors including fake pattern
- ✓ **Mail Server Existence Check:** This checks the availability of the email address domain using DNS MX records
- ✓ **Mail Existence Check:** This checks if the email address really exists and can receive email
- ✓ **Catch-All Domain Email Check:** This checks if the email domain will receive all of the email messages addressed to that domain, even if their addresses do not exist in the mail server.
- ✓ **Disposable Email Address Check:** This checks if the email is provided by a known Disposable Email Address (DEA) provider such as Mailinator, 10MinuteMail, GuerrillaMail and about 2000 more.

EMAIL VALIDATION STATUS AND STATUS CODES

Our email validation API is a web service API and uses status codes to indicate API success or errors. The status codes provide further information regarding the result of the validation and indicate why the validation of an email may have failed.

The API defines the validity of an email address as follows using only 3 statuses and each of these statuses have their corresponding status codes.

Status	Description/Meaning
Valid	Mailbox exists and not handled by Catch-all domains or known to be a DEA
Invalid	Mailbox does not exists
Unknown	Mailbox could not be verified or is determined to be handled by a Catch-all domain, DEA, Greylisted,, SMTP/Mailbox timeouts, Temporary mailbox unavailability. Specific reason for failure is provided in the status codes.

Each of these Statuses is linked to the following status Codes:

Status Codes	Meaning
Mailbox Exists and Active	The email was successfully verified as Valid
Known Disposable Email Domain	This failure means that the email address is provided by a well-known disposable email address provider (DEA) such as mailinator.com
Syntax Error	This failure means that the email is not syntactically correct
Domain Does Not Exist	This means that the email domain has been found to be non-existent
Mailbox Not Found	This failure means that the mailbox for the provided email address does not exist.
DNS Query Error	This failure means that there was a DNS error when querying the MX server
SMTP Connection Blocked	This failure means that the external mail exchanger rejected the local sender address or the incoming connecting IP.
Mailbox Validation Error	This failure means that a timeout or error occurred while verifying the existence of the mailbox for the provided email address.
Mailbox temporary not reachable (Graylisting)	This failure means that the requested mailbox is temporarily unavailable; this is not an indicator that the mailbox actually exists or not but, often, a message sent by external mail exchangers with greylisting enabled.
Mailbox Not Reachable	This failure means that the email address could not be verified because the remote server was not responding

Catchall Email Domain	This failure means that the external mail exchanger under test accepts fake, non-existent, email addresses; therefore the provided email address MAY be inexistent too. In most cases, these Catch-all domains are now setup by ISPs and ESPs as Catch-all Spam Trap domains specifically targeted to catch spammers using Dictionary Spam Attacks.
SMTP Connection Error	This failure means that a connection could not be established with the remote SMTP server
Curse Words Check	This status code indicates that the email address contained a curse word which most probably indicate it is a fake email address. E.g: fuck@yahoo.com
Fake Email Pattern Match	This status code indicates that the email address was detected to be fake using the API in-built fake email pattern detection algorithm. E.g: ususjsusjsjss@yahoo.com
Typo Checking	This status code indicates that a typo error was detected for a known email domain such as : john@hotmaill.com
InvalidToken	An invalid API key was used. Please check the API key and make sure it is correct
NoMoreQueries	The allocated # of queries or requests for the API key has been exhausted.
InternalError	There was an unexpected error on our server.
InternalDBError	This indicates that there was a database connection error from our API server

Invalid JSON Response	This error indicates that an error was received in the output of the results during the API call.
------------------------------	---

RESOURCE URL

The API URL request is a simple GET request with 3 parameters as follows:

- **API Key:** This is the unique key to authenticate to the API. Each key has a pre-allocated requests quota
- **API Server IP:** This is your dedicated API server IP that was provided to you.
- **Email:** The email address to verify

All resources are exposed according to the following URL structure:

http://<IP>/CheckEmail/<api_key>/<email_address>

To ensure data privacy and secure the key from eavesdropping, we offer the option to make the API request over HTTPS. Please note that however, HTTPS support may cause a considerable speed performance hit due to the encryption overhead.

HOW TO USE THE EMAIL VALIDATION/SCRUBBING API:

To make an API call to the Email Validation API, you will need:

1. Your Email Validation API Key (APIKey)
2. Your dedicated API server IP (IP)
3. The Email Address to check (Email_Address)

Note: The API server IP(s) will be provided in the welcome email after an API key order is made

The API call must be made using the URL format below:

http://<IP>/CheckEmail/<api_key>/<email_address>

Parameters:

- **Email_Address:** This is the email address to be verified.

Example> johndoe@yahoo.com

- **APIKey:** This is your API Key to authenticate to the API

- **IP:** This is your API server IP

Example:

http://198.58.101.227/CheckEmail /t3dxuewqjuv/johndoe@yahoo.com

AUTHENTICATION

Clients must authenticate to the API by providing their API key. Care must be taken to secure the key from unauthorized access. It is your responsibility to keep your API key secure at all times and ensure that unauthorized users do not have access to it.

HTTP STATUS CODES

Swift Email Verifier uses conventional HTTP status codes to indicate success or failure of a request. Clients must examine the HTTP status code of server replies before attempting to interpret their content. Below are the possible HTTP status codes which the API may return:

Status Code	Meaning
200 OK	The request was well formed, contains no error and the reply contains the result
404 Bad Request/Not Found	The request URL is invalid or not available
503 Service Unavailable	The API server is temporarily down. Please try the alternate URL or request for support

Note: Unless something goes wrong, the HTTP status code will always be 200 regardless whether the email is valid or invalid.

REPRESENTATIONS

The data interchange format for the API is [JSON](#). When validating server replies, clients should ensure that all expected fields are present in the responses. Each response will contain the following data.

- **Email Address:** This is the email address that was verified
- **Status:** This is the validation status for the email as provided [here](#)
- **StatusCode:** This is the corresponding status code for the email status as provided [here](#)

Some sample request responses are as follows:

```
{"Address": "lamboo7000@yahoo.com", "Status": "Valid", "StatusCode": "Success" }
```

```
{"Address": "lamboo7000@yahoyyo.com", "Status": "Invalid", "StatusCode": "DomainDoesNotExist" }
```

```
{"Address": "anonyproz@gmail.com", "Status": "Good", "StatusCode": "OK" }
```

```
{"Address": "john@gmail.com", "Status": "Bad", "StatusCode": "Bogus"}
```

```
{"Address": "service@gmail.com", "Status": "Bad", "StatusCode": "RoleAccount"}
```

API KEY REQUEST QUOTA CHECK

By default, each API key is issued with a specific limit of allowed queries or requests. If your query limit or key quota has been exhausted, you can top-up the API key with additional quota. To get information about your current request quota, you can make the following API call in the formats below:

Email Validation API Remaining Quota Check URL: `https:// server_host/path_to_resource /{APIKey}/TokenInfo`

Sample response: `{"ActualQueryCount":995,"InitalQueryCount":1000,"Status":"Ok","Token":"50000"}`

To learn more about our pricing and to top-up your API key quota, please go to our website link below:

<https://webemailverifier.com/#prices>

GETTING SUPPORT

If you have any questions or encounter any issues while implementing the API, you can reach our support center by visiting: <https://www.gondorland.com/supportsuite/>

To request for information about your current request quota, you can send us an email or support ticket. Please make sure you provide your API key when requesting for the information

To place order for API keys, please go to our website link below:

<https://www.gondorland.com/member/signup.php>

To learn more about our pricing and to top-up your API key quota, please go to our website link below:

<http://www.webemailverifier.com>