DOTS Email Validation 3

Introduction

DOTS Email Validation 3 (EV3) is a web service that provides validity and metadata information about an email address. The service provides common data elements such as syntax validity along with more refined data such as SMTP failures and deliverability flags.

EV3 can help provide instant email data verification to websites or enhancement to contact lists.

GET YOUR FREE API TRIAL KEY

If you are an existing client and are using the previous version of this service then please click on the following link.

DOTS Email Validation 2

Overview of How it Works

These are the essential checks that are performed by each operation. The service operation steps through each section to determine if an email is valid. If one step fails, subsequent checks are not made because they have been logically eliminated. Example: if the syntax on an email fails, neither the DNS nor the SMTP check is done.

Step 1: Email Correction - The email is first cleaned and corrected before being validated and verified. Extraneous text and characters are removed and common email deformities are fixed. Typos, misspellings as well as incomplete domains are also corrected.

Step 2: Syntax Check - The email is tested to verify that the format is valid, such as an "@" symbol, a domain, and no odd characters that aren't allowed in email addresses. The rules specific to the domain are also checked. For example, if you input aa@aol.com it will pass the syntax check but fail the domain specific syntax check, and therefore we do not need to continue with the DNS or SMTP level checks.

Step 3: DNS Check - The DNS or domain name check verifies that the domain exists and has a valid MX record in order to relay mail.

Step 4: SMTP Check - Once we obtain the location of the mail server from a successful DNS check, we start communicating with the target mail server. No email is actually sent to the email address being verified. This step gives us three pieces of information. It tells us if the server is working, if it will accept any address, and if will it accept this specific address. One difficulty of email validation is dealing with the defensive behaviors of email servers. Because of the growth of spam and email-mining tools, SMTP servers often respond to requests for information in cryptic and defensive ways. Mail servers may respond very slowly to information requests, provide unhelpful data, or sometimes no data at all. We have carefully crafted our system with this in mind, and therefore the data we return is still very accurate. However, sometimes we are still forced to return an unknown result for some of the SMTP level checks.

Step 5: Integrity Checks - A deliverable email address is not always a good email address. Just because an email does not bounce back does not mean that it was received, and so our service performs a variety tests and checks to evaluate the integrity of an email address. For example an email address may be a disposable address that is only temporarily deliverable or worst yet a spam trap.

Developer Guide Map

Operations

This section lists the DOTS Email Validation 3 operations and goes into the details behind the inputs and outputs.

Operations:

ValidateEmailAddress **(Recommended Operation)**

ValidateEmailFull

ValidateEmailFullNoCorrections

ValidateEmailFast

ValidateEmailFastNoCorrections

Codes, Notes, Correction

This section shows additional supporting data tables for the DPV and Correction code values returned by DOTS Email Validation operations.

Errors

This section reflects details on the error outputs that can happen with the service.

Code Snippets and Sample Code

Here you'll find code snippets for various programming languages and frameworks along with links to our sample code page on the web site.

Try The API

This is where you'll go to take the API for a spin. There you can test our recommended operation GetAddressInsight.

Service Reference

In this section you'll find all the different endpoints supported by this service, input and output schema information as well as an opportunity to try the other endpoints as well.

Frequently Asked Questions

This is a list of some of the questions we hear more often that you can reference and get answers on right away.

Integration Basics

Integrating EV3 into your application should be easy and straightforward. If you are using a common platform, such as asp, vb, C# .NET, PHP and others, Service Objects may already have sample code built that you can use:

https://www.serviceobjects.com/developers/sample-code/

However, if you are using a common platform that does not already have sample code, you can ask Service Objects to build an example for you. Email support@serviceobjects.com for more details.

Web Service Structure

Web services provide a standard interface to encapsulate tricky business logic. They allow simple integration of applications via the web. Service Objects has followed web services best practices and come up with some of its own standards to ensure that its web services are as easy to integrate, and as accessible as possible.

The host path, or physical location of the web service is here:
https://trial.serviceobjects.com/ev3/web.svc

A test page for the web service can be found here:
https://trial.serviceobjects.com/ev3/

The location of the WSDL, or Web Service Definition Language document, is here (This is also accessible via the "Service Definition" link.):
https://trial.serviceobjects.com/ev3/soap.svc?wsdl

Important Note!
SOAP is done via POST, only with special XML markup in the post-body.

The WSDL is an XML document that defines the interaction web service, meaning its inputs, outputs, operations, and the like. Most likely, you will have another tool read this WSDL and make the operations available to you in your application via some type of proxy class. Whenever your utilities or IDE asks for a WSDL path, you can provide this one. Every web service has *operations* that it offers to subscribers. These operations, also called methods, contain different functionality and return different outputs.