Email Address Format and Syntax Explained

Posted on November 26, 2021

Have you ever wondered what it means when people say that email addresses should follow the standard set by the Internet Engineering Task Force (IETF), the organization tasked to set up standards for Internet usage? Simply put, it has to do with email syntax. In this post, we'll tackle what the right email syntax or format is, what email syntax errors are, and how email syntax validation can help reduce issues.

What Is Email Syntax?

Email syntax means following the format the IETF set for email addresses so these can be considered valid, meaning they can be reached and receive messages.

All email addresses should have an email prefix and an email domain. The email prefix (also known as "local") appears before the @ symbol. It usually contains the recipient's name (e.g., jane, janesmith, jane_smith, jane-smith, or jane.smith) or a specific department's name in an organization (e.g., sales, accounting, or contact). The email domain, meanwhile, appears after the @ symbol. It usually contains the company name and the top-level domain (TLD) that the organization uses (e.g., .com, .co, or .org). The email domain is usually the company domain, as in microsoft[.]com, google[.]com, or amazon[.]com. Apart from periods, dashes, and underscores, no other special characters can be added to email prefixes and domains.

That said, an example of a properly formatted email address or one that follows the proper syntax would be janesmith@microsoft[.]com, where "janesmith" is the email prefix and "microsoft[.]com" is the email domain.

Examples of Email Syntax Errors

We provided a sample email syntax error above but there are loads more. The following table shows some examples of valid and invalid email prefixes and domains for clarity.

Invalid Email Prefixes	Valid Email Prefixes
name.@mail.com	**firstname.lastname**@mail.com
firstname..lastname @mail.com	**firstname.lastname**@mail.com
.name@mail.com	**name**@mail.com

Invalid Email Domains	Valid Email Domains
firstname.lastname@**mail.c**	firstname.lastname@**mail.com**
firstname.lastname@**mail#archive.com**	firstname.lastname@**mail-archive.com**
firstname.lastname@**mail**	firstname.lastname@**mail.co**

From the sample email syntax errors in the tables above, take note of the following pointers:

- An email prefix can only be up to 64 characters long. The characters can be a combination of any of the 26 letters of the English alphabet, the numbers 0–9, and special characters limited to an exclamation point (!), the sharp symbol (#), a dollar sign ($), the percentage symbol (%), an ampersand (&), an apostrophe ('), an asterisk (*), the plus sign (+), a hyphen (–), an open slash mark (/), the equal sign (=), a question mark (?), a circumflex or caret (^), an underscore (_), a period (.), brackets ({ or }), a vertical bar (|), or a tilde mark (~). Note, however, that a period can only be used once in an email prefix. It can't appear as the first or last character as well.

- An email domain can only be up to 255 characters long. The characters can be any alphanumeric combination, including non-Latin characters in the case of internationalized domain names (IDNs), and hyphens. Email domains should always end with a period followed by a valid TLD. Finally, hyphens can't appear as the first or last character in an email domain.

Why Should Email Addresses Follow the Required Email Syntax?

The reason for adhering to the correct email address format is simple: Without following proper email syntax, senders can’t send messages to intended recipients’ inboxes because email services won’t know where to direct emails.

On the sender’s part, that translates to a hard bounce that can negatively affect its email sending reputation and hurt its overall domain reputation. If that problem persists, the sender could end up on the spam blocklists that email service providers (ESPs) and Internet service providers maintain.

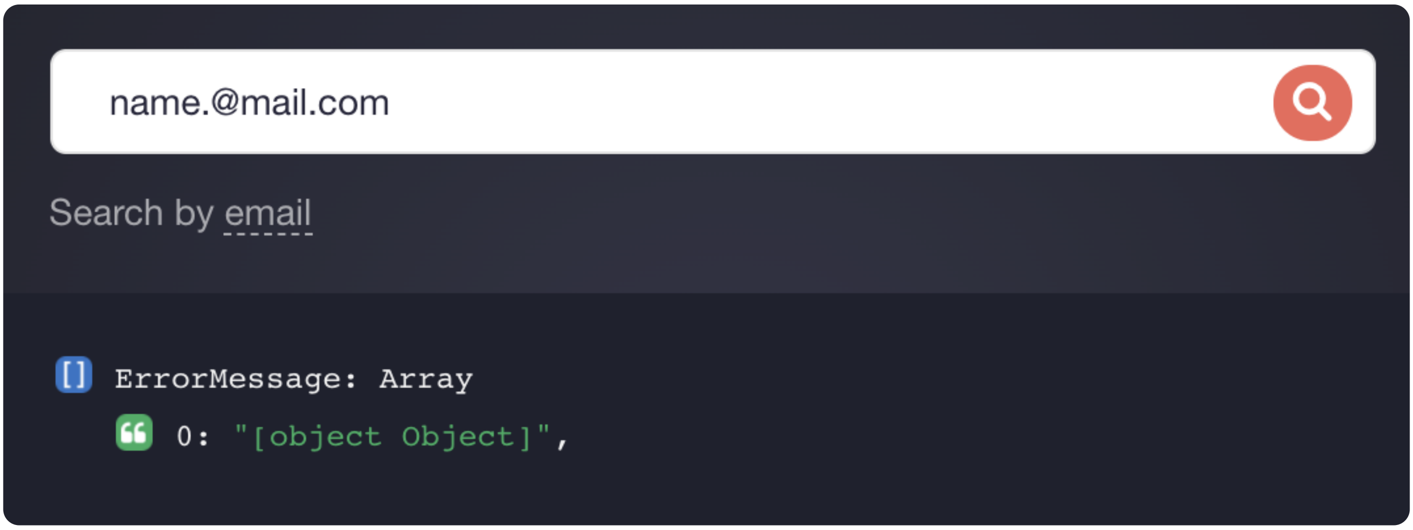
On the recipient's end, meanwhile, that could mean missing out on important business transactions or leads.

How Do You Validate Email Syntax?

While validating email syntax can be done manually, this approach is rather time-consuming and prone to error. A far easier way to detect invalid email address formats is to use an email verification tool.

Organizations can integrate an email verification API into their signup forms so users with invalid email addresses due to improper formatting can be automatically disallowed from registering for or accessing any of their offerings. They can also use the tool to get rid of invalid email addresses from their distribution lists. Let's see the results using the sample invalid email addresses we identified earlier.

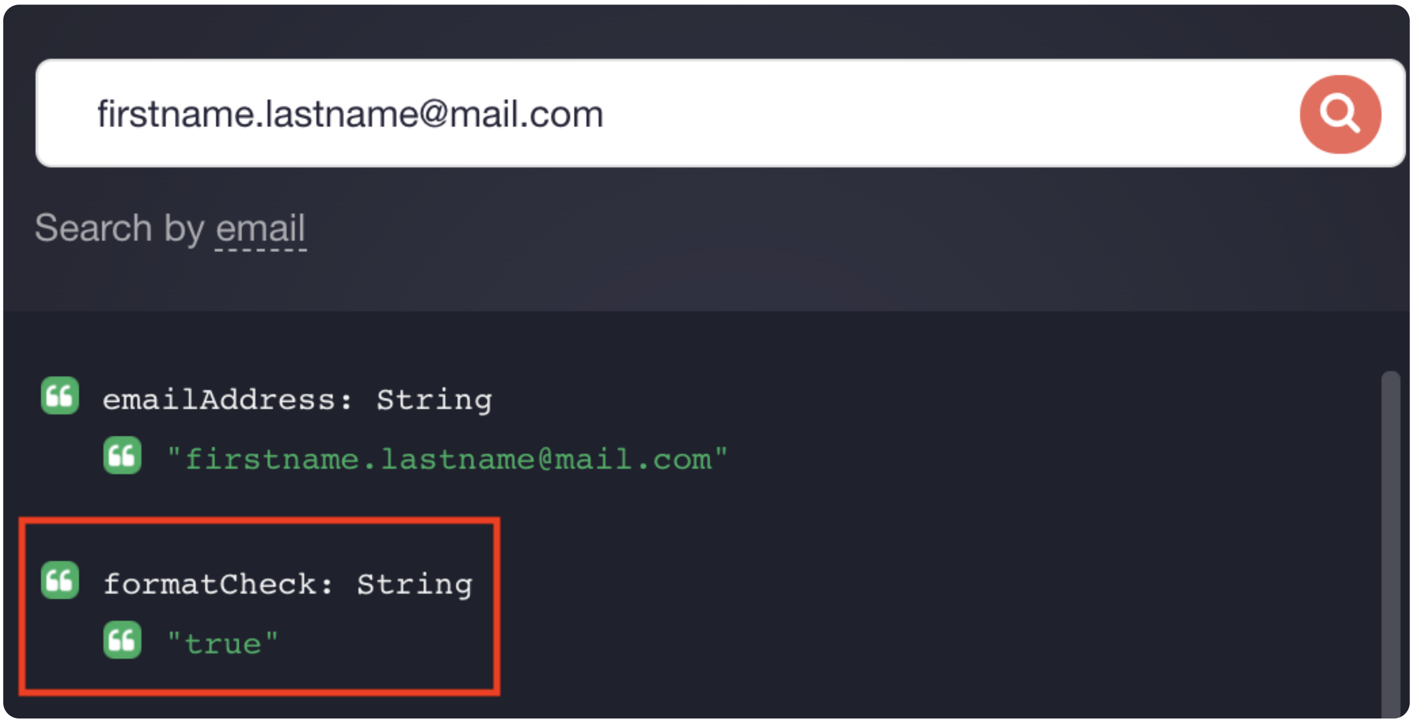
We queried the email address **name.@mail.com** (with an invalid email prefix) on Email Verification API and got this result:



name.@mail.com

```
[] ErrorMessage: Array
    0: "[object Object]",
```

Here is the result for its valid email address equivalent:



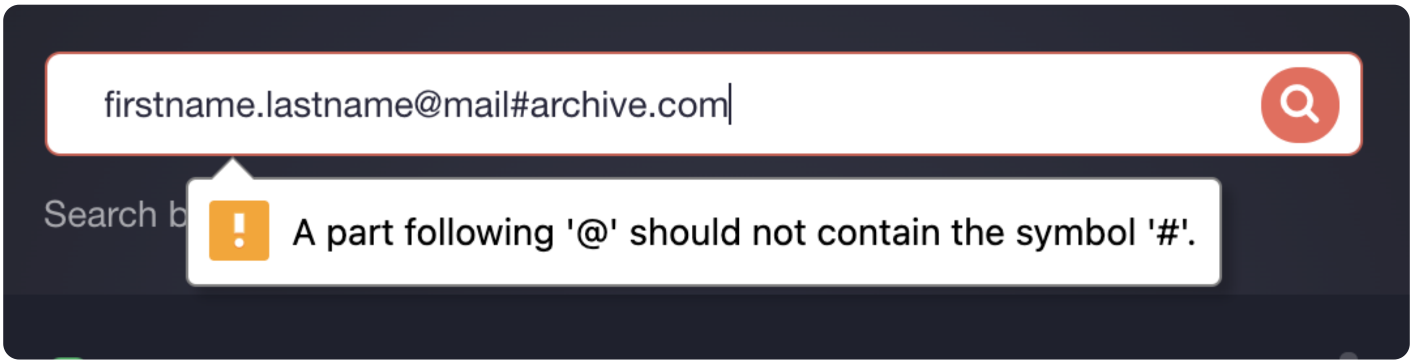
firstname.lastname@mail.com

Search by email

```
emailAddress: String
  "firstname.lastname@mail.com"

formatCheck: String
  "true"
```

We also queried the email address **firstname.lastname@mail#archive.com** (with an invalid email domain) on the tool and got this result:

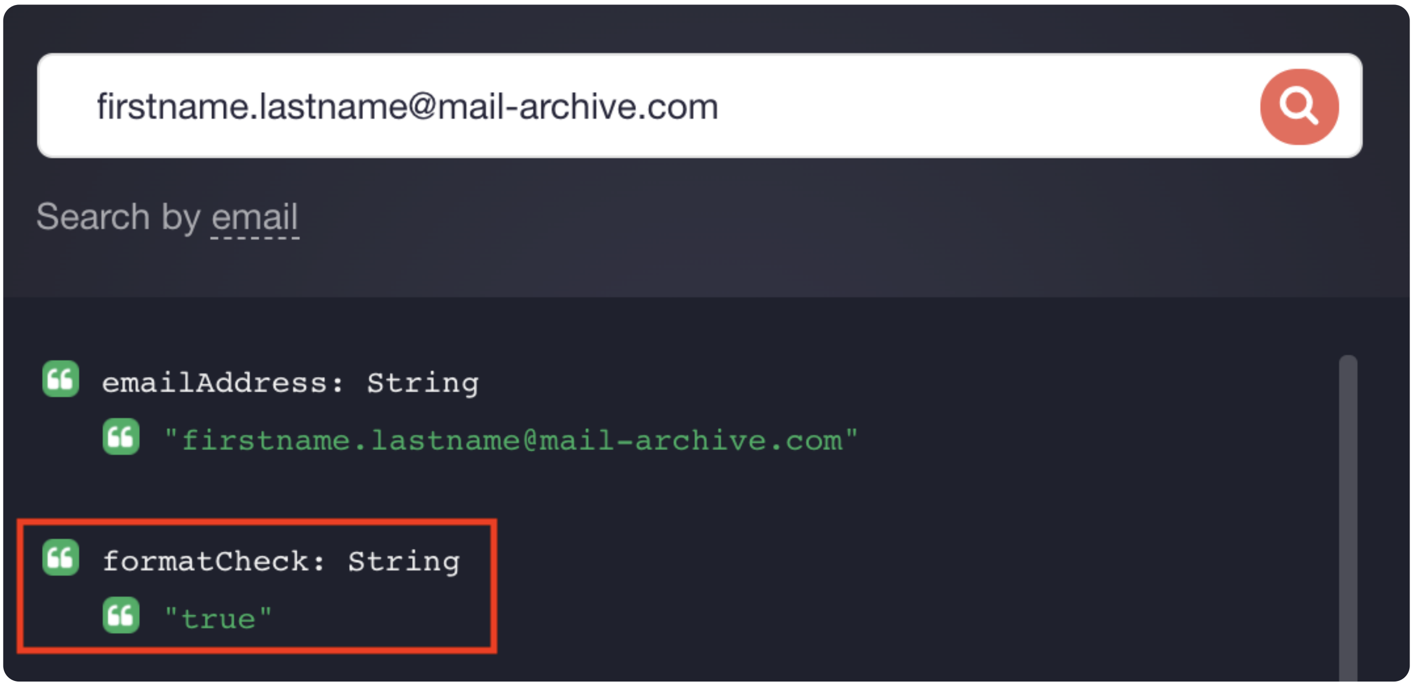


firstname.lastname@mail#archive.com

Search b

A part following '@' should not contain the symbol '#'.

Here is the result for its valid email address equivalent:



firstname.lastname@mail-archive.com

Search by email

```
emailAddress: String
    "firstname.lastname@mail-archive.com"

formatCheck: String
    "true"
```

As the Email Verification API results show, integrating the tool into signup forms reduces the chances that a company’s distribution list would fill up with invalid email addresses due to syntax.

Improper email syntax can affect the quality of leads that organizations get from their signup, registration, and resource gating forms. With the aid of an email verification tool, however, they can be assured of disallowing invalid email address formats from the get-go.