|  |
| --- |
| **Email Validation API** |

|  |
| --- |
| **Team members:** 1. Akshit  2. Janvi Kumari    **Start Date for API** 13/05/2023 |

**Expected completion date for API**

29/05/2023

**Estimated revised date** 31/05/2023

|  |
| --- |
| **Overview:** The Email Validation API is a RESTful API that provides an easy way to validate email addresses. It determines whether the email is authentic, discarded, and whether the domain is valid or not. It can also recognise typical role-based email accounts like admin, sales, and support. For a variety of applications, the API offers a quick and trustworthy means of email validation. |

|  |
| --- |
| **API Functionality and Use Cases** |
| **Functionality**: The API accepts an email address as input and verifies it using a number of criteria, such as syntax, domain reachability and existence, and whether or not it is disposable.  Additionally, it determines the email address's function based on frequent role keywords in the username, like "info" or "support."  The API can also show the email address's reputation based on whether it has been reported as spam or not. |
| **Use Cases**:  **1)** **Email validation**: Before sending an email to an address, users can use the API to check that it is valid and capable of receiving messages.  **2) Filtering of disposable email addresses**: Users can use the API to exclude addresses that are frequently used for spam and fraud.  **3)** **Email routing based on roles**: Users can use the API to send emails to particular email addresses according to their roles, for example, sending all "sales" emails to a particular email address.  **4)** **Monitoring email reputation**: Users can use the API to keep track of an email address's reputation and take the necessary action if it is reported as suspicious or malicious. |

**Out of scope** :

1) Email delivery or sending emails.

2) Email content analysis, including spam detection or phishing analysis.

3) Email tracking or analytics.

4) Bulk email verification or validation.

5) Any other features not directly related to email validation.

Output will be in Text format.

|  |  |
| --- | --- |
| **Functional Requirements** | |
| **Input** | Email address to be validated in the JSON format in the request body. |
| **Operation** | The API will perform the following operations on the entered email address:   * Verify if the email address is syntactically valid. * Verify if the domain of the email address exists and is valid. * Check if the email address is disposable or not. * Determine if the email address belongs to any common roles like info, sales, support etc. * Determine the reputation of the email address. |
| **Output** | The API will return the following data in JSON format in the response body:   * The input email address. * A Boolean value indicating whether the email address is valid or not. * The domain of the email address. * A string indicating if the email exixts or not. * A string indicating the role of the email address or personal if it doesn't belong to any common role. * A string indicating the reputation of the email address (Spam , Not Spam). |

**Examples of API Quotes Functionality:**

**API parameters:**

|  |  |
| --- | --- |
| **Parameters** | **Type Details** |
| Email | A string representing the email address to be validated.  The email parameter should be a valid email address in the standard format of **username@domain.com**  The type for email parameter is string. |

**Response:**

|  |  |
| --- | --- |
| **‘**email**’** | The input email address. |
| ‘valid’ | Boolean indicating whether the email is valid or not. |
| ‘domain’ | The domain part of the email address, if the email is valid. |
| ‘exists’ | A string indicating whether the email exists or not. |
| ‘role’ | The role of the email address, if it contains one of the predefined role keywords. |
| ‘reputation’ | A string indicating the reputation of the email address, which can be: ‘Spam’ , ‘Not Spam’ |

**Response and error codes:**

|  |  |  |
| --- | --- | --- |
| **Code** | **Type** | **Details** |
| 200 | OK | Everything worked as expected |
| 400 | Bad request | Bad requests |

**Dependency:**

A session on Github and swagger documentation.

|  |  |
| --- | --- |
| **Quotes Timeline(Details of Milestones)** | |
| Quotes Specification Document | 17/05/2023(Tentative) |
| Environment and Tool Preparation | 18/05/2023(Tentative) |
| Coding | 19/05/2023(Tentative) |
| Unit Testing | 25/05/2023(Tentative) |
| Acceptance Testing | 29/05/2023(Tentative) |

A session on Github and swagger documentation.

**Languages and Tools to be used:**

Python, VScode, GitHub

**Development and Test Environment:**

● Local machine

**Expected Hosting Details and End points:**

Host: Exponentialhost.com

Input and output endpoint names will be decided at the deployment stage.

**Documentation Plan:**

<swagger documentation>

**Acceptance Testing:**

Input quality:

● Input shall not be in the form of any other language other than English.

● Max limit of the text should not be exceeded.

Output quality:

● Empty output should not be returned.

**Future work/potential:**

* Adding support for bulk email validation, allowing users to validate a large list of email addresses at once.
* Implementing a feedback loop to improve the accuracy of the email validation algorithm over time.
* It could be integrated with anti-spam services or threat intelligence platforms to provide a more complete picture of the email address being validated.