# HSBC Python Developer Coding Test

Design and implement a RESTful API service using any Python framework you wish. The service will accept IPv4 addresses and domains, query external services and return appropriate responses.

## You Will Need

* A python IDE
* Python version 3.8+
* A URLScan.io API Key
  + These are free and can be obtained here: <https://urlscan.io/about-api/>
    - If you do not wish to sign up for an API key, please let us know and we will create one for you
* A VirusTotal API key
  + These are free and can be obtained here: <https://support.virustotal.com/hc/en-us/articles/115002088769-Please-give-me-an-API-key>
    - If you do not wish to sign up for an API key, please let us know and we will create one for you

Input

The service should expose 3 endpoints:

|  |  |  |
| --- | --- | --- |
| METHOD | SOURCE | INPUT |
| GET | PATH | type: str = “ip” or “domain”  data: <ipv4 or domain name>  Example:   * /ip/1.2.3.4, * /domain/hsbc.com |
| GET | PARAMS | type: str = “ip” or “domain”  data: <ipv4 or domain name>  Example:   * ?type=ip&data=1.2.3.4 * ?type=domain&data=hsbc.com |
| POST | BODY (JSON) | type: str = “ip” or “domain”  data: <ipv4 or domain name>  Example:   * {“type”: “ip”, “data”: “1.2.3.4”} * {“type”: “domain”, “data”: “hsbc.com”} |

## Processing

Incoming data should be parsed to ensure it ***looks*** valid. Either data modelling or regular expressions are acceptable for this.

When data is validated, you should perform the following actions:

* Submit the data to both VirusTotal and URLScan.io
* Parse the data and determine whether either of these services consider the data malicious

## Output

You should return data according to this schema with an appropriate status code:

{

“virustotal”: Union[dict[virustotal\_object], None],

“urlscan”: Union[dict[urlscan\_object], None],

“malicious”: bool

}

## Extra Credit

* Resolve domains and include the resolved IP in the return object
* Include additional free or opensource APIs/services to query

## Caveats

* Feel free to use existing libraries or spin your own
* The service should run as-is either directly or via a service manager like Gunicorn