DNSBLv5APIv3: updateEntry (PUT)

Usage

Endpoint

/3.0/dnsbl

Description

Update or create a new entry in the DNS Blacklist.

Syntax

The format of this PUT request can be made in several ways. The base look of the data you send is this:

Standard request

```
{
    "ip":["44.11.12.77"]
}
```

Just as in the regular lookup case DNSBLv5APIv3: getListed (POST).

Variants

Be careful with the variant described here. Putting data into the system **do** support multiple addresses. This however, changes the requirements of the input data object. In the old API, adding data without a bitmasked value would set the bitmask value 64 to the host (**IP_ABUSE_NO_SMTP**). In the APIv3 service empty bitmasks on multiple hosts are not allowed, since we prefer to get proper values on each added host (in case they are different to each other). So, this does not work:

Multiple hosts (not allowed)

```
{
    "ip":["44.11.12.77","18.33.14.30"]
}
```

This is what happens in such cases:

Rendered error on multiple hosts without bitmasks

```
{
    "response": [],
    "errors": {
        "code": "400",
        "success": "",
        "faultstring": "Updating or adding multiple entries requires a syntax with associative arrays (arrays
with keys)"
    }
}
```

Request of adding/updating multiple hosts

To be more specific with what kind of address you're adding, you can also format your input syntax like this:

Adding with associative arrays - Simple host

```
{
        "ip":{"44.11.12.77":"32"}
}
```

Sending this data into the DNSBL, will add the IP-address **44.11.12.77** with the bitmask value of **32**.

In fact, this kind of syntax therefore supports adding multiple hosts in one call:

Adding with associative arrays - Multiple hosts

```
{
        "ip": {
                "44.11.12.77":"32",
                "18.33.14.30":"104"
        }
}
```

In this call, you're adding the first host with the flag **IP_SECOND_EXIT** and the second with multiple flags (102 = **IP_ABUSE_NO_SMTP**, **IP_SECOND_EXI**
, **IP_CONFIRMED** and **IP_PHISHING**).

When this document is written, the output response is not completed yet - so the below example might change over time.

DNSBL Response

```
{
        "dnsblResponse": {
                "status": [{
                                "address": "44.11.12.77",
                                "arpa": "77.12.11.44",
                                "state": "new",
                                "arpaDelegations": [
                                        "77.12.11.44.dnsbl.tornevall.org"
                                ],
                                "flag": "32"
                        },
                        {
                                "address": "18.33.14.30",
                                "arpa": "30.14.33.18",
                                "state": "new",
                                "arpaDelegations": [
                                        "30.14.33.18.dnsbl.tornevall.org",
                                        "30.14.33.18.bl.fraudbl.org"
                                ],
                                "flag": "102"
                        }
                ]
        }
}
```

States

States in the response output can be **new** and **update** (if the state is "new" - the host has been added - and "updated" for hosts that already exists in the database).

The update-state might affect the hit ratio of the blacklist. This means that if someone recently requested for removal in the system and the host are again discovered by the DNSBL, a penalty timer can be added to the restored host. In this way, hosts that is popular spam targets, renders a high risk making it harder to get delisted in the future. Unless the administrators gets the problems solved.