



# Web Security On-prem

v8.5.x

Management API Guide

© 2025 Forcepoint

Forcepoint and the FORCEPOINT logo are trademarks of Forcepoint.  
All other trademarks used in this document are the property of their respective owners.

Published 07 January 2025

Every effort has been made to ensure the accuracy of this document. However, Forcepoint makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. Forcepoint shall not be liable for any error or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information in this documentation is subject to change without notice.

# Contents

<b>1 Management API</b> .....	5
Introduction.....	5
Management API overview.....	6
Management API object types.....	7
<b>2 Interfaces</b> .....	11
Introduction.....	11
Start a Management API transaction.....	12
Add API-managed categories.....	13
Add URLs and IP addresses to an API-managed category.....	14
Delete API-managed categories.....	17
Delete URLs and IP addresses from an API-managed category.....	18
List API-managed categories or all categories.....	21
List URLs and IP addresses in an API-managed category.....	23
Commit all categories, URLs, and IP addresses.....	25
Roll back all category, URL, and IP address additions and deletions.....	25
Get system and transaction status.....	26
Examples using curl.....	27
<b>3 v8.5.xApiParameters.ini</b> .....	31
How to make changes to the file.....	31
Configurable parameters.....	31



## Chapter 1

# Management API

### Contents

- [Introduction](#) on page 5
- [Management API overview](#) on page 6
- [Management API object types](#) on page 7

## Introduction

---

The Management API for Forcepoint Web Security is a REST interface using JSON. It allows administrators to:

- Create custom, API-managed categories
- Add URLs and IP addresses to API-managed categories
- Remove URLs and IP addresses from API-managed categories
- Delete API-managed categories
- View categories, URLs, IP addresses, and API status

See *Management API object types*, for more information about the objects (categories, URLs, and IP addresses) that can be managed via the API.

API-managed categories:

- Appear in category lists in the Forcepoint Security Manager, marked with the tag **(restricted)**
  - The “(restricted)” tag indicates that URLs and IP addresses within the category are managed exclusively through the API.
  - Administrators cannot use the Security Manager to add recategorized URLs or keywords to API-managed categories.
- Are automatically assigned to the risk class **Security Risk**
- Can be assigned actions (block, permit, and so on) in category filters
- Appear in reports, alerts, and SIEM logs
- Are **not** used by the hybrid service  
Users whose traffic is sometimes managed by on-premises components and sometimes managed by the hybrid service may therefore see differences in policy enforcement, depending on whether they are inside or outside of your network.

If you have not yet deployed the Management API, see the [Management API Deployment & Installation Guide](#) for instructions.


To start using the Management API, see:

- *Management API overview*, for an introduction to API transactions and client/server interaction.
- *Management API object types*, for an overview of the object types that the API can manipulate.
- *Interfaces*, for information about the methods supported by the API.
- *v8.5.xApiParameters.ini*, for information about configuring API behavior.

**Related concepts**[Management API object types](#) on page 7[Management API overview](#) on page 6**Related information**[Interfaces](#) on page 11[v8.5.xApiParameters.ini](#) on page 31

# Management API overview

The process of using the Management API works as follows:

- 1) A client starts a transaction.
  - 2) The client receives a transaction ID.
  - 3) The client issues one or more commands, using the transaction ID.
  - 4) The client completes the transaction in one of 2 ways:
    - Commits the transaction  
This causes all commands issued within the transaction to be completed.
- 
- Note**
- While the commit returns an immediate response, the commands and data that are part of the transaction continue to be processed in the background.
- You can start a new transaction immediately after issuing the commit command, even though data from the previous transaction is still being processed.
- Rolls back the transaction  
This causes all commands issued within the transaction to be discarded.

To prevent data collisions, the Management API allows only one transaction at a time. Once a transaction is started, no other transactions can start until the first transaction has been committed or rolled back. This is true no matter how many instances of the Policy API Server component have been deployed.

If a connection is interrupted before a transaction is committed or rolled back, the transaction times out in 10 minutes, by default. The transaction timeout period can be configured in the **ApiParameters.ini** file (see *ApiParameters.ini*).

**Related information**[v8.5.xApiParameters.ini](#) on page 31

# Communication between clients and the Management API

There are 2 options when a client connects to the Management API:

- In deployments where there are limited concerns about the security of traffic between the client and the Management API machine, the client can connect using only basic authentication. For example:

```
curl -k -u username:password https://10.82.5.74:15873/web/api/v1/categories
```
- In deployments where there is more concern about securing traffic between client and server:
  - 1) Copy the server certificate to a directory on the client system that can be accessed by the administrator communicating with the API.
  - 2) Specify both the certificate name and the basic authentication account when connecting to the Management API server. For example:

```
curl --cacert cert_name.crt -u username:password https://10.203.30.40:15873/web/api/v1/categories
```

In either case, if the connection account password is not specified in the request, the administrator is prompted to provide it.

Instructions for defining the account used for basic authentication, as well as instructions for creating the server certificate, can be found in the [Management API Deployment & Installation Guide](#).

## Management API object types

There are 3 object types recognized by the Management API.

### Related concepts

[Categories](#) on page 7

[URLs](#) on page 8

[IP addresses](#) on page 9

## Categories

When you start using the Management API, the first step is to create an API-managed category and obtain its unique category ID.

- The category must be identified by **name** or **ID** when adding URLs and IP addresses.
- You can add URLs and IP addresses only to API-managed categories. An attempt to use another category (that is, a Master Database category or custom category created via the Forcepoint Security Manager) is considered an error, which causes the request to fail.

API-managed categories may include the following attributes:

- **Name** (required): When creating an API-managed category, you must give it a unique, alphanumeric name.
  - The name cannot contain any of the following characters:

\* < > { } ~ ! \$ % & @ # . " | \ & + = ? / ; : ,

- The category name and ID must be unique within the deployment. An API- managed category cannot be given the same name as an existing Forcepoint- defined or custom category.
- The category name appears in the Forcepoint Security Manager and in reports.
- You may specify either the category name or ID (see below) to add, delete, modify, or view URLs and IP addresses.
- **ID** (required): When you create an API-managed category, it is automatically assigned a unique ID.
  - The ID is a number with a value of 1899 or higher.
  - After committing a transaction, use the **list categories** command to find each new API-managed category's unique ID (see *List API-managed categories or all categories*).
- **Description** (optional): The category description, if any, appears both in the Security Manager and in the response to the API call used to view the category. It may contain alphanumeric characters, periods, and commas. It cannot include any of the following characters:
  - \* < > { } ~ ! \$ % & @ # . " | \ & + = ? / ; :
- **Level** (optional): When you add an API-managed category, you can also define a parent category.
  - If the parent category does not exist, it is added along with the child category.
  - If not otherwise defined, the parent category is assigned the default value of **0** (Miscellaneous). Up to two levels of subcategory may be added under Miscellaneous. For example:

*Miscellaneous*

*First\_Level\_Child\_Category*

*Second\_Level\_Child\_Category*

The Miscellaneous category is a Forcepoint-defined category that cannot be changed.

If administrators attempt to add URLs or IP addresses to anything other than an API- managed category, a 400 (bad request) error is returned. This indicates that the category name or ID is invalid because it:

- Does not exist
- Belongs to a Forcepoint-defined category
- Belongs to a custom category defined via the Security Manager

### Related reference

[List API-managed categories or all categories](#) on page 21

## URLs

URLs in the system are defined according to [RFC 3986](#).

- Only the hostname field (part of the authority) is required.
- Other parts are optional, but can be used to define a stricter match.
- CGI parameters (anything after the “?” in a URL) are automatically removed from the URL before it is saved in the Management API local database.
- If no protocol is specified, the Management API automatically adds all 3 of the following to its database: http://, https://, and ftp://.

Use Punycode to add URLs containing non-ASCII characters to API-managed categories.

URLs can be added to more than one API-managed category with multiple calls to the **add** methods (see *Add URLs and IP addresses to an API-managed category*). When the URL is matched, all categories for the URL are returned for use in policy enforcement.

**Related reference**

[Add URLs and IP addresses to an API-managed category](#) on page 14

## IP addresses

---

IP addresses and ranges are as specified by [RFC-791](#) (IPv4) and [RFC 4291](#) (IPv6).

IP addresses and ranges can be added as an element under an API-managed category. The destination IP address for a URL request is checked against the IP addresses and ranges added via the API. If the IP address is matched, the category containing that IP address is added to the list of categories detected by the product.

IP addresses and ranges can be added to more than one API-managed category with multiple calls to the **add** methods (see *Add URLs and IP addresses to an API-managed category*). When an IP address is matched, all categories for the IP address are returned for use in policy enforcement.

**Related reference**

[Add URLs and IP addresses to an API-managed category](#) on page 14



## Chapter 2

# Interfaces

### Contents

- Introduction on page 11
- Start a Management API transaction on page 12
- Add API-managed categories on page 13
- Add URLs and IP addresses to an API-managed category on page 14
- Delete API-managed categories on page 17
- Delete URLs and IP addresses from an API-managed category on page 18
- List API-managed categories or all categories on page 21
- List URLs and IP addresses in an API-managed category on page 23
- Commit all categories, URLs, and IP addresses on page 25
- Roll back all category, URL, and IP address additions and deletions on page 25
- Get system and transaction status on page 26
- Examples using curl on page 27

## Introduction

The HTTPS methods mentioned in this chapter are used to add, delete, and list categories, URLs, and IP addresses via the Management API.

In all of the commands described in this document, **<ps\_ip\_address>** is the IP address of the Policy Server machine hosting the Management API. Note that changes made via any single Management API instance are stored to Policy Broker and replicated to all Management API instances within a short period.

Syntax examples for each the commands can be found at the end of this chapter; see *Examples using curl*.

In addition, a series of sample Python scripts for issuing Management API commands is available for download. Edit the sample scripts or use them for reference. See:

[http://www.websense.com/content/support/library/web/v84/mgmt\\_api\\_guide/api\\_example\\_code.tar](http://www.websense.com/content/support/library/web/v84/mgmt_api_guide/api_example_code.tar)

### Related concepts

Examples using curl on page 27

# Start a Management API transaction

Use the POST command described in this section to start a session for adding and deleting categories, URLs, and IP addresses via the Management API. The `start` command must be run before any PUT, POST, or DELETE commands.

Only one Policy API Server in a global Forcepoint Web Security deployment can start a transaction at a time. If a transaction is in progress and administrators attempt to initiate a new transaction, a 409 error is returned.

Method	URL	Description
POST	<code>https://&lt;ps_ip_address&gt;:15873/ api/web/v1/categories/start</code>	Starts a transaction for adding and deleting categories, URLs, and IP addresses

After performing subsequent PUT, POST, or DELETE commands, run a `commit` command to commit the changes to the system (see *Commit all categories, URLs, and IP addresses*). If you do not want the transaction to be committed, use the `rollback` command to stop the transaction and discard all of the commands that are part of the transaction (see *Roll back all category, URL, and IP address additions and deletions*).

All subsequent requests after the transaction is started must include a transaction ID. If the ID is not included, a 400 (bad request) error is returned.

## Related reference

[Commit all categories, URLs, and IP addresses on page 25](#)

[Roll back all category, URL, and IP address additions and deletions on page 25](#)

## JSON response

The response includes the transaction ID, which can be used for issuing subsequent commands, including commit or rollback.

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0"
}
```

## Response codes

200	OK
400	Bad request
403	Authentication failed
409	Conflict (another transaction is in progress)
500	Internal server error

# Add API-managed categories

The POST command described in this section can be used to add categories as containers for URLs and IP addresses.

Method	URL	Description
POST	https://<ps_ip_address>:15873/ api/web/v1/categories	Adds categories as described in a JSON request

## JSON request data

The JSON request includes:

- The **transaction ID** obtained in the **start** transaction call
- The **category name** (required; must be unique)  
An error is returned if the category name is not unique.
- The **category description** (optional; blank by default)
- The ID of the **parent** category
  - **0** is the top-level category, and can be specified as the parent.
  - Categories can be added either to Forcepoint-defined (Master Database) categories or to API-managed categories. They cannot be added to custom categories created via the Forcepoint Security Manager.
  - Subcategories can be added to categories that have a parent of 0, but not to categories with a different parent.

For example:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Categories":
  [
    {
      "Category Name": "Malicious XYZ",
      "Category Description" : "Categories found by XYZ engine",
      "Parent": 0
    },
    {
      "Category Name": "Malicious ABC",
      "Category Description" : "Categories found by ABC engine",
      "Parent" : 128
    }
  ]
}
```

## JSON response to add categories request

The JSON response includes the API-managed category names added to the transaction. For example:

```
{
  "Categories":
  [
    {"Category Name": "Malicious XYZ"},
    {"Category Name": "Malicious ABC"}
  ]
}
```

## Response codes

200	OK (categories added successfully)
400	Bad JSON format or data
403	Authentication failed
409	Bad transaction ID or not in a transaction
500	Internal server error

## JSON for bad request

```
{
  "Error" :
  [
    "Bad JSON format.",
    "Not in a transaction",
    "Other Error Message"
  ]
}
```



### Note

Category names and IDs are not checked until the data is committed to the system. Error messages for duplicate and missing categories (by name or ID) appear in the response to a **status** API call after the **commit** has finished.

## Add URLs and IP addresses to an API-managed category

The POST command described in this section can be used to add URLs, IP addresses, and ranges to an API-managed category.

Method	URL	Description
POST	https:<ps_ip_address>:15873/ api/web/v1/categories/urls	Adds new URLs or IP addresses to the API-managed category specified in the JSON request

## JSON request data

### Adding URLs and IP addresses by category name

When the API-managed category for a set of URLs or IP addresses is specified by name, the JSON request includes:

- The **transaction ID** obtained in the **start** transaction call
- The **category name** (required; must be unique)  
This field is used to define which category is returned when a URL or IP address is matched. An error is returned if the category name does not correspond to the ID of an API-managed category already in the system.
- The **URLs** to add to the category (optional if IP addresses are present)
- The **IPs** (IP addresses or ranges) to add to the category (optional if URLs are present)

For example:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category Name": "Malicious XYZ",
  "URLs":
  [
    "http://www.test1.com/test1",
    "http://www.test1.com/test2",
    "http://www.test1.com/test3"
  ],
  "IPs":
  [
    "167.34.15.117",
    "167.34.16.118-167.34.16.120",
    "2045:1221:1231::1331"
    "24.56.8.0/23",
    "55AF:F451::/32"
  ]
}
```

### Adding URLs and IP addresses by category ID

- The **transaction ID** obtained in the **start** transaction call
- The **category ID**  
This field is used to define which category is returned when a URL or IP address is matched. An error is returned if the category ID does not match the ID of an API-managed category already in the system.  
The field can appear as part of the file path in the URI or in the JSON request.
- The **URLs** to add to the category (optional if IP addresses are present)

- The **IPs** (IP addresses or ranges) to add to the category (optional if URLs are present)

For example:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category ID": 1945,
  "URLs":
  [
    "http://www.test1.com/test1",
    "http://www.test1.com/test2",
    "http://www.test1.com/test3"
  ],
  "IPs":
  [
    "167.34.15.117",
    "167.34.16.118-167.34.16.120",
    "2045:1221:1231::1331",
    "24.56.8.0/23",
    "55AF:F451::/32"
  ]
}
```

## JSON response to adding URLs and IP addresses to categories

The **Categories** field contains an array of API-managed category IDs and the total number of URLs and IP addresses added to each category.

```
{
  "Categories":
  [
    {
      "Name": "Malicious XYZ",
      "ID": 1945,
      "Totals":
      {
        "Added URLs": 3,
        "Added IPs": 5
      }
    }
  ]
}
```

## Response codes

<b>200</b>	OK (URLs and IP addresses successfully added to the transaction)
<b>400</b>	Bad JSON format or data
<b>403</b>	Authentication failed
<b>409</b>	Bad transaction ID or not in a transaction
<b>500</b>	Internal server error

## JSON for bad request

```
{
  "Error" :
  [
    "Not in a transaction",
    "Other Error Message"
  ]
}
```

## Delete API-managed categories

The DELETE and POST commands described in this section can be used to delete API-managed categories.

Method	URL	Description
DELETE	https://<ps_ip_address>:15873/ api/web/v1/categories	Deletes categories as specified in the JSON request
POST	https:<ps_ip_address>:15873/ api/web/v1/categories/delete	Alternative format for curl and Python, which do not accept a payload for the DELETE method

## JSON request data

### Delete categories by ID

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category IDs":
  [
    1987,
    2078
  ]
}
```

### Delete categories by name

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category Name":
  [
    "Malicious ABC"
  ]
}
```

## Response codes

200	OK (category and its URLs and IP addresses deleted)
403	Authentication failed
409	Bad transaction ID or not in a transaction
500	Internal server error

## JSON for bad request

```
{
  "Error" :
  [
    "Not in a transaction",
    "Other Error Message"
  ]
}
```



### Note

Category names and IDs are not checked until the data is committed to the system. Error messages for duplicate and missing categories (by name or ID) appear in the response to a **status** API call after the **commit** has finished.

## Delete URLs and IP addresses from an API-managed category

This topic explains how to remove URLs, IP addresses, or IP ranges from a specific API-managed category using API commands. DELETE and POST are two HTTP methods available for this operation.

To remove URLs, IP addresses, and ranges from a specific API-managed category:

Method	URL	Description
DELETE	https:<ps_ip_address>:15873/api/web/v1/categories/urls	Deletes URLs and IP addresses from an API-managed category, as specified in the JSON request.
POST	https:<ps_ip_address>:15873/api/web/v1/categories/delete/urls	Alternative format for curl and Python, which do not accept a payload for the DELETE method.



### Note

If you have installed Instrumental Build (IB) for policy API version 8.5.5 or later, use the DELETE method mentioned below to remove URL and IP address. The POST method remains same.

Method	URL	Description
DELETE	https://<ps_ip_address>:15873/api/web/v1/categories/urls?Transaction ID=0871fada-a592-11ef-8e13-b3947b515934&Category Name=<category name>&URLs=<urls1>,<urls2>	Deletes URLs and IP addresses from an API-managed category, as specified in the Query string request.

Example for DELETE method:

```
"URL":
https://10.203.201.31:15873/api/web/v1/categories/urls?Transaction
ID=0871fada-a592-11ef-8e13-b3947b515934&Category Name=Malicious XYZQ&URLs=https://
www.newwrl.com,https://www.newwrl.com

"IP":
https://10.203.201.31:15873/api/web/v1/categories/urls?Transaction
ID=0871fada-a592-11ef-8e13-b3947b515934&Category Name=Malicious XYZQ&IPs=167.34.15.117
```

## JSON request

The JSON request has a list of URLs, IP addresses, or both to delete from the given category.

- If a URL, IP address, or range is not found, processing continues to the next URL, IP address, or range. The total number of items deleted of each type reflects the actual number removed from the system, and not any skipped items.
- If a URL, IP address, or range has another category associated with it, the item remains in that second category, but is still included in the count of deleted items (since it was removed from the first category).
- The total number of items deleted can be higher than expected if duplicates exist and are removed.

Example 1:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category ID": 1945,
  "URLs":
  [
    "http://www.test1.com/test1",
    "http://www.test1.com/test2",
    "http://www.test1.com/test3"
  ],
  "IPs":
  [
    "167.34.15.117",
    "167.34.16.118-167.34.16.120",
    "2045:1221:1231::1331",
    "24.56.8.0/23",
    "55AF:F451::/32"
  ]
}
```

## Example 2:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category Name": "Malicious_10_05_2017",
  "URLs":
  [
    "http://www.test1.com/test1",
    "http://www.test1.com/test2",
    "http://www.test1.com/test3"
  ],
  "IPs":
  [
    "167.34.15.117",
    "167.34.16.118-167.34.16.120",
    "2045:1221:1231::1331",
    "24.56.8.0/23",
    "55AF:F451::/32"
  ]
}
```

You can use the \* wildcard to delete **all** custom URLs and IP addresses from a specific category. The category itself is not deleted.

**Important**

While you can use "\*" to delete all URLs and IP addresses from an API-managed category, other uses of wildcards and regular expressions are **not** supported.

## Example 3:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category ID": 2156,
  "URLs": ["*"],
  "IPs": ["*"]
}
```

## Example 4:

```
{
  "Transaction ID": "44081c08-453a-11e6-a9de-d99213c464e0",
  "Category Name": "Malicious_01_01_2016",
  "URLs": ["*"],
  "IPs": ["*"]
}
```

## JSON response for deleting custom URLs and IP addresses from a category

```
{
  "Category Name": "Malicious_01_01_2016",
  "Category ID": 2156,
  "Deleted ":
  {
    "Deleted URLs": 2,
    "Deleted IPs": 5
  }
}
```

## Response codes

200	OK (URLs and IP addresses deleted)
400	Bad JSON format or data
403	Authentication failed
409	Bad transaction ID or not in a transaction
500	Internal server error

## JSON for bad request

```
{
  "Error" :
  [
    "Not in a transaction",
    "Other Error Messages"
  ]
}
```



### Note

Category names and IDs are not checked until the data is committed to the system. Error messages for duplicate and missing categories (by name or ID) appear in the response to a **status** API call after the **commit** has finished.

## List API-managed categories or all categories

The following GET command can be used to view a list of either:

- API-managed categories
- **All** categories (including custom categories created in the Forcepoint Security Manager and via the API)  
This option is provided to give administrators a way to find the ID for Master Database or custom categories that will be used as a parent for one or more API- managed categories.

The Management API cannot use these IDs in any other way (for example, it is not possible to delete URLs from a Master Database category).

If a transaction ID is not provided, the category list is limited to the categories stored in the system. If a transaction ID is provided, any new categories in the pending transaction are included.

Method	URL	Description
GET	https:<ps_ip_address>:15873/api/web/v1/categories/all	Lists all categories (including Forcepoint-defined Master Database categories)

Method	URL	Description
GET	https://<ps_ip_address>:15873/api/web/v1/categories	Lists all API-managed categories
GET	https://<ps_ip_address>:15873/api/web/v1/categories?transactionid=<txn_id>	Lists all categories, including categories that have not yet been committed

## JSON response for listing categories

Each entry has the parent and its category ID, followed by any children and their category IDs.

```
{
  "Categories":
  [
    {
      "Category Name": "User-Defined",
      "Category ID":64,
      "Category Description":"User defined categories",
      "Category Hierarchy":45000,
      "Category Owner":"Forcepoint",
      "Children":
      [
        {
          "Category Name": "Malicious XYZ",
          "Category ID":1910,
          "Category Description":"Malicious URLs from XYZ",
          "Category Hierarchy":65535,
          "Category Owner":"API"
        },
        {
          "Category Name": "Malicious ABC",
          "Category ID":1911,
          "Category Description":"Malicious URLs from ABC",
          "Category Hierarchy":65535,
          "Category Owner":"API"
        },
        {
          "Category Name": "Malicious DEF",
          "Category ID":1912,
          "Category Description":"Malicious URLs from DEF",
          "Category Hierarchy":65535,
          "Category Owner":"API"
        }
      ]
    },
    {
      "Category Name": "Nudity",
      "Category ID":65,
      "Category Description":"User defined categories",
      "Category Hierarchy":44000,
      "Category Owner":"Forcepoint"
    }
  ]
}
```

## Response codes

200	OK (request successful)
400	Bad request
403	Authentication failed
500	Internal server error

## JSON for bad request

```
{
  "Error" :
  [
    "Unable to access category data",
    "Other Error Messages"
  ]
}
```

## List URLs and IP addresses in an API-managed category

Use the GET command described in this section to view a list of all committed URLs, IP addresses, and ranges in an API-managed category. URLs and IP addresses in a pending transaction are not returned with this command.

Method	URL	Description
GET	https:<ps_ip_address>:15873/ api/web/v1/categories/ urls? catname=<category_name>  <i>or</i> https:<ps_ip_address>:15873/ api/web/v1/categories/ urls? catid=<category_id>	Lists all URLs, IP addresses, and ranges in the specified category.  The category may be specified by name or ID.

# JSON response for viewing URLs and IP addresses

Each entry has the category name and ID, followed by a list of URLs, then a list of IP addresses and ranges.

```
{
  "Category Name": "Malicious DEF",
  "Category ID": 1945,
  "URLs": [
    "http://www.test1.com/test1",
    "http://www.test1.com/test2",
    "http://www.test1.com/test3"
  ],
  "IPs": [
    "167.34.15.117",
    "167.34.16.118-167.34.16.120",
    "2045:1221:1231::1331",
    "24.56.8.0/23",
    "55AF:F451::/32"
  ]
}
```

## Response codes

200	OK (request succeeded)
400	Bad request
403	Authentication failed
409	Bad category ID or name
500	Internal server error

## JSON for bad request

```
{
  "Error" : [
    "Unable to access URL data",
    "Other Error Messages"
  ]
}
```

# Commit all categories, URLs, and IP addresses

The POST command described in this section records API-managed categories, URLs, IP addresses, and ranges. Once committed, they can be used in policy enforcement and reporting. The call is non-blocking and returns immediately. Use the status call (*Get system and transaction status*) to determine when the commit is complete.

Method	URL	Description
POST	https:<ps_ip_address>:15873/api/web/v1/categories/commit?transactionid=<id>	Commit the API transaction

## Related reference

[Get system and transaction status](#) on page 26

## JSON response for commit

The response includes commit completion time.

```
{
  "Transaction ID":"a468af86-453a-11e6-88af-a180af187bfd",
  "Commit Time":"April 8, 2016 11:37 am"
}
```

## Response codes

200	OK (request succeeded)
400	Bad request
403	Authentication failed
404	Transaction not found
500	Internal server error

# Roll back all category, URL, and IP address additions and deletions

Use the POST command described in this section to cancel all commands included in the specified transaction.

Method	URL	Description
POST	https:<ps_ip_address>:15873/api/web/v1/categories/rollback?transactionid=<id>	Cancels all add and delete commands within the specified transaction

## JSON response for rollback

The response includes the ID for the transaction that was rolled back.

```
{
  "Transaction ID": "a468af86-453a-11e6-88af-a180af187bfd",
  "Rollback Time": "April 8, 2016 11:37 am"
}
```

## Response codes

200	OK (request succeeded)
400	Bad request
403	Authentication failed
404	Transaction not found
500	Internal server error

## Get system and transaction status

The GET command described in this section returns status information that describes the health of the system, as well as the total number of URLs, IP addresses, and ranges in the system. The status request does not require a transaction ID.

Method	URL	Description
GET	https:<ps_ip_address>:15873/api/web/v1/categories/status	Returns: <ul style="list-style-type: none"> <li>Overall health of the system</li> <li>Total number of categories, URLs, and IP addresses</li> <li>Error summaries</li> </ul>

## JSON response to a status request

```
{
  "Build Status" : "Done",
  "Cat Engine Health" : "OK",
  "HttpServer Health" : "OK",
  "Number of bad requests" : 0,
  "Number of good requests" : 19,
  "Number of unauthorized accesses" : 0,
  "Number requesting bad paths" : 0,
  "Status" : [
    "Deleted category: Api_Cat_106 (name)",
    "Database cdb1477694574.db created at 10-28-2016
    22:42:54.720233 for transaction "
  ],
  "Total API-managed categories from last call" : 10,
  "Total IP addresses from last call" : 48,
  "Total URLs from last call" : 254,
  "Total categories from last call" : 0,
  "Total connection errors" : 0,
  "Total data fetch errors" : 3,
  "Total errors" : 0,
  "Total locked transaction errors" : 0,
  "Total non-specialized Errors" : 0,
  "Total requests received" : 19,
  "Total transaction errors" : 0
}
```

## Examples using curl

In addition to the examples below, you can download a tar file containing a series of sample Python scripts:

[https://help.forcepoint.com/websec/en-us/on-prem/85/mgmt\\_api\\_guide/api\\_example\\_code.tar](https://help.forcepoint.com/websec/en-us/on-prem/85/mgmt_api_guide/api_example_code.tar)

The sample scripts demonstrate how administrators might automate use of the Management API.

## Adding API-managed categories, URLs, and IP addresses

Start a new transaction:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/start
```

Add an API-managed category (all data in the HTTPS request):

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories -d
"{\"Transaction
ID\": \"<transaction_ID_string>\", \"Categories\": [
{ \"Category Name\": \"<name_string>\", \"Category
Description\": \"<description_string>\", \"Parent\":
<numeric_category_ID>}]}"
```

Add API-managed categories (data in a JSON file):

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories -d
@<filename>.json --header "Content-Type: application/json"
```

Or:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories --upload-file /
<path>/<filename>.json
```

Add a URL (all data in the HTTPS request):

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/urls -d
"{\"Transaction ID\": \"<transaction_ID_string>\", \"Category
ID\": <numeric_ID>, \"URLs\": [\"https://www.new_url.com/\"]}"
--cacert PolApiServer.crt
```

Add URLs or IP addresses (data in a JSON file):

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/urls -d
@<filename>.json --header "Content-Type: application/json"
```

Or:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/urls --upload-
file /<path>/<filename>.json
```

Commit the transaction:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/
commit?TransactionID="<id_string>"
```

## Deleting API-managed categories, URLs, and IP addresses

Start a new transaction:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/start
```

Delete an API-managed category:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/delete -d
"{\"Transaction ID\": \"<transaction_ID_string>\", \"Category
Names\": [\"<category_name>\"]}"
```

Or:

```
curl -k -u <username>:<password> -X POST https://
<ps_ip_address>:15873/api/web/v1/categories/delete -d
"{\"Transaction ID\": \"<transaction_ID_string>\", \"Category
IDs\": [\"<category_ID>\"]}"
```

Delete URLs or IP addresses from an API-managed category:

```
curl -k -u <username>:<password> -X POST https:  
<ps_ip_address>:15873/api/web/v1/categories/delete/ urls -d  
@<filename>.json --header "Content-Type: application/json"
```

Or:

```
curl -k -u <username>:<password> -X POST https:  
<ps_ip_address>:15873/api/web/v1/categories/delete/ urls --  
upload-file /<path>/<filename>.json
```

Commit the transaction:

```
curl -k -u <username>:<password> -X POST https://  
<ps_ip_address>:15873/api/web/v1/categories/  
commit?TransactionID="<id_string>"
```



# v8.5.xApiParameters.ini

### Contents

- How to make changes to the file on page 31
- Configurable parameters on page 31

## How to make changes to the file

The **ApiParameters.ini** file can be used to configure several aspects of Management API behavior. The file is located in the `/opt/Websense/bin` directory on the Policy API Server machine.

### Steps

- 1) Open the file in a text editor.
- 2) Edit the parameter or parameters that you want to change.
- 3) Save and close the file.
- 4) Use the `/opt/Websense/WebsenseDaemonControl` script to restart the **PolicyAPIServer** daemon.

## Configurable parameters

The configurable parameters in the file are:

Parameter	Default Value	Description
RestServerCertPath	.	Directory for storing server certificate files. By default, the Policy API Server installation directory: <code>/opt/Websense/bin</code>
RestServerCertRoot	PolApiServer.crt	Name of the server certificate file
RestServerCertKey	PolApiServer.key	Name of the server certificate key file
ApiLoadTimeoutSec	30	Timeout period for loading the API database (in seconds)

Parameter	Default Value	Description
CDatabasePath	/opt/Websense/bin/	Directory that contains the local database for API-managed categories, URLs, and IP addresses. Must be a fully-qualified path.
TransactionTimeoutMin	10	Timeout period for API transactions (in minutes)

Other parameters in the file should not be changed unless you are advised to do so by Forcepoint Technical Support.

