blacklistManagement

Service Description

Abstract
This document provides service description for the **blacklistManagement** service.

Contents

1 Overview

This document describes the **blacklistManagement** service, which enables systems (with operator role or proper permissions) to handle (query, create, remove) blacklist entries in bulk. An example of this interaction is when an operator bans a consumer because it is flooding the Local Cloud with thousands of requests. To enable other systems to use, to consume it, this service needs to be offered through the ServiceRegistry.

The **blacklistManagement** service contains the following operations:

- *query* lists the entries that match the filtering requirements;
- *create* adds the specified entries to the blacklist;
- *remove* deactivates the entries that apply to the systems with the specified names.

The rest of this document is organized as follows. In Section 2, we describe the abstract message operations provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned operations.

1.1 How This Service Is Meant to Be Used

The service's purpose is to handle the blacklist entries centrally and in bulk. Thus, the **blacklistManagement** service makes it possible to manage who is (not) banned from the Local Cloud.

Application systems should not use this service, only human operators (indirectly via some tool, for example) or dedicated Support systems.

1.2 Important Delimitations

The requester has to identify itself to use any of the operations.

1.3 Access policy

The service is only available for operators, dedicated Support systems and those who have the proper authorization rights to consume it.

2 Service Operations

This section describes the abstract signatures of each operations of the service. The **blacklistManagement** service is used to *query*, *create* and *remove* blacklist entries. In particular, each subsection names an operation, an input type and one or two output types (unsuccessful operations can return different structure), in that order. The input type is named inside parentheses, while the output type is preceded by a colon. If the operation has two output types, they are separated by a slash. Input and output types are only denoted when accepted or returned, respectively, by the operation in question. All abstract data types named in this section are defined in Section 3.

2.1 operation query (BlacklistQueryRequest) : BlacklistEntryListResponse / ErrorResponse

Operation *query* lists the blacklist entries that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- If a filter expects a list, there is an OR relation between the elements of the filter.
- There is an AND relation between different kind of filters.
- If *alivesAt* is set, inactive records will not be returned.

2.2 operation create (BlacklistCreateListRequest) : BlacklistEntryListResponse / ErrorResponse

Operation *create* creates the specified blacklist entries. The creation data must meet the following criteria:

- System names are case sensitive, must follow the PascalCase naming convention and have to be unique within the Local Cloud.
- System names can contain maximum 63 character of letters (English alphabet), and numbers, and have to start with a letter.
- The expiration date has to be in the future.
- Reason must always be specified and can contain maximum 1024 characters.

2.3 operation remove (BlacklistRemoveRequest) : OperationStatus / ErrorResponse

Operation *remove* deactivates the entries that apply to the specified system names. The entries will stay in the database and it is possible to query them later as inactive records.

3 Information Model

Here, all data objects that can be part of the **blacklistManagement** service are listed and must be respected by the hosting system. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.9, which are used to represent things like hashes and identifiers.

3.1 struct BlacklistQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.
pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
systemNames	List<SystemName>	no	Requester is looking for blacklist entries that apply to systems with any of the specified names.
mode	Mode	no	Requester is looking for blacklist entries with the specified activity.
issuers	List<SystemName>	no	Requester is looking for blacklist entries that were created by systems with any of the specified names.
revokers	List<SystemName>	no	Requester is looking for blacklist entries that were revoked by systems with any of the specified names.
reason	String	no	Requester is looking for blacklist entries that were created for the specified reason or the reason contains this text.
alivesAt	DateTime	no	Requester is looking for active blacklist records that are not expired at this timestamp.

3.2 struct Identity

An Object which describes the identity of a system. It also contains whether the identified system has higher level administrative rights.

3.3 struct BlacklistEntryListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<BlacklistEntryResponse>	List of blacklist entry results.
count	Number	The total number of corresponding entries.

3.4 struct BlacklistEntryResponse

Field	Type	Description
systemName	SystemName	Unique identifier of the blacklisted system.
createdBy	SystemName	Unique identifier of the system that created the record.
revokedBy	SystemName	Unique identifier of the system that revoked the record. Only appears if the record was revoked.
createdAt	DateTime	Blacklist record was created at this timestamp.
updatedAt	DateTime	Blacklist record was updated at this timestamp.
reason	String	The system was blacklisted because of this reason.
expiresAt	DateTime	Blacklist record expires at this timestamp. Only appears if the record can expire.
active	Boolean	Indicates if the rule defined by the entry is active. Only false if the rule has been explicitly revoked.

3.5 struct ErrorResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
errorMessage	String	Description of the error.
errorCode	Number	Numerical code of the error.
type	ErrorType	Type of the error.
origin	String	Origin of the error.

3.6 struct BlacklistCreateListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
entries	List<BlacklistCreateRequest>	yes	List of blacklist entries to create.

3.7 struct BlacklistCreateRequest

Field	Type	Mandatory	Description
systemName	SystemName	yes	The name of the system to be blacklisted.
expiresAt	DateTime	no	This rule will expire at this timestamp.
reason	String	yes	The system is blacklisted because of this reason.

3.8 struct BlacklistRemoveRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
systemNames	List<SystemName>	yes	The names of the systems to remove from the blacklist.

3.9 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
Boolean	One out of true or false.
DateTime	Pinpoints a specific moment in time.
Direction	The direction of a sorting operation. Possible values are the representation of ascending or descending order.
ErrorType	Any suitable type chosen by the implementor of service.
List<A>	An *array* of a known number of items, each having type A.
Mode	Specifies whether the queried records should have the *active* flag set. The possible values are: ALL, ACTIVES, INACTIVES.
Number	Decimal number.
Object	Set of primitives and possible further objects.
OperationStatus	Logical, textual or numerical value that indicates whether an operation is a success or a failure. Multiple values can be used for success and error cases to give additional information about the nature of the result.
String	A chain of characters.
SystemName	A string identifier that is intended to be both human and machine-readable. Must follow PascalCase naming convention.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.0.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.0.0	