

Resources

Page Contents

- [Active Connector](#)
- [Assertion Security Zone](#)
- [Bundle](#)
- [Cassandra Connection](#)
- [Certificate](#)
- [Cluster Property](#)
- [Custom Key Value Store](#)
- [Email Listener](#)
- [Encapsulated Assertion](#)
- [Firewall Rule](#)
- [Folder](#)
- [Generic Entity](#)
- [Group](#)
- [Http Configuration](#)
- [Identity Provider](#)
- [Interface Tag](#)
- [JDBC Connection](#)
- [JMS Destination](#)
- [Listen Port](#)
- [Policy](#)
- [Policy Alias](#)
- [Policy Bundle Installer](#)
- [Private Key](#)
- [Published Service](#)
- [Resource Document](#)
- [Revocation Checking Policy](#)
- [Role](#)
- [Sample Message](#)
- [Secure Password](#)
- [Security Zone](#)
- [Service Alias](#)
- [Site Minder Configuration](#)
- [User](#)

Active Connector

[Top](#)

Active connectors are connectors that poll for messages. For example MQ Native Queues and SFTP Polling Listeners are examples of Active Connectors.

Create Or Update

Creates or Updates an existing active connector. If an active connector with the

given ID does not exist one will be created, otherwise the existing one will be updated.

Request

```
PUT 1.0/activeConnectors/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the active connector to create or update

Body

Active connector to create or update

Element	I7:ActiveConnector
Content-Type	application/xml

Response

Body

A reference to the newly created or updated active connector.

Delete

Deletes an existing active connector.

Request

```
DELETE 1.0/activeConnectors/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the active connector to delete.

Response

No Response Body

Get

Returns an active connector with the given ID.

Request

```
GET 1.0/activeConnectors/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the active connector to return

Response

Body

The active connector.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example active connector that can be used as a reference for what active connector objects should look like.

Request

GET 1.0/activeConnectors/template

Response

Body

The template active connector.

Element	l7:Item
----------------	---------

Create

Creates a new active connector.

Request

POST 1.0/activeConnectors

Body

The active connector to create

Element	l7:ActiveConnector
Content-Type	application/xml

Response

Body

A reference to the newly created active connector

List

Returns a list of active connectors. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/activeConnectors?name=MySFTPPollingListener
```

Returns active connector with name "MySFTPPollingListener".

```
/restman/1.0/activeConnectors?type=SFTP&name=MySFTPPollingListener&name=MyOtherSFTPPollingLis
```

Returns active connector of SFTP type with name either "MySFTPPollingListener" or "MyOtherSFTPPollingListener"

If a parameter is not a valid search value a bad request error will be returned.

Request

```
GET 1.0/activeConnectors
```

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
enabled	boolean			Enabled filter
type	string			Type filter
hardwiredServiceId	string			Service ID filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of active connectors. If the list is empty then no entities were found.

Element	l7:List
----------------	---------

Assertion Security Zone

Assertion Security Zones entities are used to apply security zones to assertions. By [Top](#)

default, assertions do not have a security zone assigned to them.

Get

Returns the assertion security zone for the assertion with the given fully qualified name.

Request

GET 1.0/assertionSecurityZones/{name}

Path Parameters

Param	Type	Description
name	string	Fully qualified name of the assertion

Response

Body

Assertion security zone for the assertion

Element	I7:Item
----------------	---------

Update

Updates an assertion security zone

Request

PUT 1.0/assertionSecurityZones/{name}

Path Parameters

Param	Type	Description
name	string	Fully qualified name of the assertion

Body

Updated assertion security zone for an assertion

Element	I7:AssertionSecurityZone
Content-Type	application/xml

Response

Body

A reference to the newly updated assertion security zone.

Template

Returns a template, which is an example assertion security zone that can be used as a reference for what assertion security zone objects should look like.

Request

```
GET 1.0/assertionSecurityZones/template
```

Response

Body

The template assertion security zone.

Element	
l7:Item	

List

Returns a list of assertion security zones. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/assertionSecurityZones?name=com.17tech.external.assertions.jdbcquery.JdbcQueryAs
```

Returns assertion security zone of the Jdbc Query Assertion. The name of the assertion security zone is the fully qualified name of the assertion.

```
/restman/1.0/assertionSecurityZones?securityZone.id=0e028eafc5c66c3af755a2e470734948
```

Returns assertion security zones that have security zone ID "0e028eafc5c66c3af755a2e470734948"

If a parameter is not a valid search value a bad request error will be returned.

Request

```
GET 1.0/assertionSecurityZones
```

Query Parameters

Param	Type	Default	Since	Description
sort	name securityZone.id			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter

securityZone.id	string			Security zone ID filter. To list all assertions with no security zones applied use the default ID: 0000000000000000ffffffffffffff
-----------------	--------	--	--	---

Response

Body

List of assertion security zones. If the list is empty then no assertion security zones were found.

Element	I7:List
----------------	---------

Bundle

[Top](#)

This resource is used to export and import bundles for migration. See [migration.html](#) for more documentation on migration and bundling.

Folder, Service or Policy Export

Returns the bundle for the given resource type. The resource type is either a policy, service, or folder

Request

```
GET 1.0/bundle/{resourceType}/{id}
```

Path Parameters

Param	Type	Description
resourceType	folder policy service	Resource type. Either folder, service or policy
id	string	ID of the resource to bundle

Query Parameters

Param	Type	Default	Since	Description
defaultAction	NewOrExisting NewOrUpdate	NewOrExisting		Default bundling action. By default this is NewOrExisting
defaultMapBy	id name guid	id		Default map by action.
includeRequestFolder	boolean	false		For a folder export, specifies whether to include the folder in the

				bundle or just its contents.
exportGatewayRestManagementService	boolean	false		If true the gateway management service will be exported too. False by default.
includeDependencies	boolean	false	1.0.1	True to export with dependencies. False by default
encryptSecrets	boolean	false	1.0.1	True to export with encrypted secrets. False by default.
encryptUsingClusterPassphrase	boolean	false	1.0.1	True to use the cluster passphrase if encrypting secrets. False by default

Header Parameters

Param	Type	Default	Since	Description
L7-key-passphrase	string		1.0.1	The optional base-64 encoded passphrase to use for the encryption key when encrypting secrets.

Response

Body

The bundle for the resource

Element	7:Item
----------------	--------

Export Bundle

Returns the bundle for the given resources. This API call is capable of returning a bundle created from multiple resources.

Request

GET 1.0/bundle

Query Parameters

--	--	--	--	--

Param	Type	Default	Since	Description
defaultAction	NewOrExisting NewOrUpdate	NewOrExisting		Default bundling action. By default this is NewOrExisting
exportGatewayRestManagementService	boolean	false		If true the gateway management service will be exported too. False by default.
folder	string			Folders to export
service	string			Services to export
policy	string			Policies to export
all	boolean	false	1.0.1	True to export the full gateway. False by default
includeDependencies	boolean	false	1.0.1	True to export with dependencies. False by default
encryptSecrets	boolean	false	1.0.1	True to export with encrypted secrets. False by default.
encryptUsingClusterPassphrase	boolean	false	1.0.1	True to use the cluster passphrase if encrypting secrets. False by default.

Header Parameters

Param	Type	Default	Since	Description
L7-key-passphrase	string		1.0.1	The optional base-64 encoded passphrase to use for the encryption key when encrypting secrets.

Response

Body

The bundle for the resources

--	--

Element	I7:Item
----------------	---------

Import Bundle

This will import a bundle.

Request

PUT 1.0/bundle

Query Parameters

Param	Type	Default	Since	Description
test	boolean	false		If true the bundle import will be tested no changes will be made to the gateway
activate	boolean	true		False to not activate the updated services and policies.
versionComment	string			The comment to set for updated/created services and policies

Header Parameters

Param	Type	Default	Since	Description
L7-key-passphrase	string		1.0.1	The optional base-64 encoded passphrase to use for the encryption key when encrypting passwords.

Body

The bundle to import

Element	I7:Bundle
Content-Type	application/xml

Response

Body

The mappings performed during the bundle import

Cassandra Connection

[Top](#)

Since: 1.0.1

Cassandra connections allow the Gateway to query external databases and then use the query results during policy consumption. Cassandra Connection

Create Or Update

Creates or Updates an existing Cassandra connection. If a Cassandra connection with the given ID does not exist one will be created, otherwise the existing one will

be updated.

Request

```
PUT 1.0/cassandraConnections/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the Cassandra connection to create or update

Body

Cassandra connection to create or update

Element	I7:CassandraConnection
Content-Type	application/xml

Response

Body

A reference to the newly created or updated Cassandra connection.

Delete

Deletes an existing Cassandra connection.

Request

```
DELETE 1.0/cassandraConnections/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the Cassandra connection to delete.

Response

No Response Body

Get

Returns a Cassandra connection with the given ID.

Request

```
GET 1.0/cassandraConnections/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the Cassandra connection to return

Response

Body

The Cassandra connection.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example Cassandra connection that can be used as a reference for what Cassandra connection objects should look like.

Request

GET 1.0/cassandraConnections/template

Response

Body

The template Cassandra connection.

Element	l7:Item
----------------	---------

Create

Creates a new Cassandra connection

Request

POST 1.0/cassandraConnections

Body

The Cassandra connection to create

Element	l7:CassandraConnection
Content-Type	application/xml

Response

Body

A reference to the newly created Cassandra connection

List

Returns a list of Cassandra connections. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/cassandraConnections?name=MyCassandraConnection
```

Returns Cassandra connection with name "MyCassandraConnection".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/cassandraConnections

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
keyspace	string			Keyspace filter
contactPoint	string			Contact point filter
port	string			Port filter
username	string			User name filter
compression	string			Compression filter
ssl	boolean			SSL filter
enabled	boolean			Enabled filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of Cassandra connections. If the list is empty then no Cassandra connections were found.

Element	l7:List
----------------	---------

Certificate

Certificates are either HTTPS and LDAPS certificates. This resource enables the management of those certificates.

[Top](#)

Create Or Update

Creates or Updates an existing certificate. If a certificate with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

```
PUT 1.0/trustedCertificates/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the certificate to create or update

Body

Certificate to create or update

Element	I7:TrustedCertificate
Content-Type	application/xml

Response

Body

A reference to the newly created or updated certificate.

Delete

Deletes an existing certificate.

Request

```
DELETE 1.0/trustedCertificates/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the certificate to delete.

Response

No Response Body

Get

Retrieves a certificate.

Request

GET 1.0/trustedCertificates/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the certificate to return

Response

Body

The certificate

Element	I7:Item
----------------	---------

Template

Returns a template, which is an example certificate that can be used as a reference for what certificate objects should look like.

Request

GET 1.0/trustedCertificates/template

Response

Body

The template certificate.

Element	I7:Item
----------------	---------

Create

Creates a new certificate

Request

POST 1.0/trustedCertificates

Body

The certificate to create

Element	I7:TrustedCertificate
Content-Type	application/xml

Response

Body

A reference to the newly created certificate

List

Returns a list of certificates. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/trustedCertificates?name=MyCertificate
```

Returns certificates with name "MyCertificate".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/trustedCertificates

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of certificates. If the list is empty then no certificates were found.

Element	7:List
----------------	--------

Cluster Property

[Top](#)

Cluster properties are used to set global properties. Example cluster properties include "cluster.hostname" and "log.levels"

Create Or Update

Creates or Updates an existing cluster property. If a cluster property with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

```
PUT 1.0/clusterProperties/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the cluster property to create or update

Body

Cluster property to create or update

Element	I7:ClusterProperty
Content-Type	application/xml

Response

Body

A reference to the newly created or updated cluster property.

Delete

Deletes an existing cluster property.

Request

```
DELETE 1.0/clusterProperties/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the cluster property to delete.

Response

No Response Body

Get

Returns a cluster property with the given ID.

Request

```
GET 1.0/clusterProperties/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the cluster property to return

Response

Body

The cluster property.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example cluster property that can be used as a reference for what cluster property objects should look like.

Request

GET 1.0/clusterProperties/template

Response

Body

The template cluster property.

Element	l7:Item
----------------	---------

Create

Creates a new cluster property.

Request

POST 1.0/clusterProperties

Body

The cluster property to create

Element	l7:ClusterProperty
Content-Type	application/xml

Response

Body

a reference to the newly created cluster property

List

Returns a list of cluster properties. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/clusterProperties?name=MyProperty
```

Returns cluster property with name "MyProperty".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/clusterProperties

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter

Response

Body

A list of cluster properties. If the list is empty then no cluster properties were found.

Element	I7:List
----------------	---------

Custom Key Value Store

[Top](#)

A Custom key value is an item that can be stored by a custom assertion.

Create

Creates a new custom key value.

Request

POST 1.0/customKeyValues

Body

The custom key value to create

Element	I7:CustomKeyValue
----------------	-------------------

Content-Type	application/xml
---------------------	-----------------

Response

Body

A reference to the newly created custom key value

Create Or Update

Creates or Updates an existing custom key value. If a custom key value with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/customKeyValues/{id}

Path Parameters

Param	Type	Description
id	string	ID of the custom key value to create or update

Body

Custom key value to create or update

Element	I7:CustomKeyValue
Content-Type	application/xml

Response

Body

A reference to the newly created or updated custom key value.

Delete

Deletes an existing custom key value.

Request

DELETE 1.0/customKeyValues/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the custom key value to delete.

Response

No Response Body

Get

Returns a custom key value with the given ID.

Request

```
GET 1.0/customKeyValues/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the custom key value to return

Response

Body

The custom key value.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example custom key value that can be used as a reference for what custom key value objects should look like.

Request

```
GET 1.0/customKeyValues/template
```

Response

Body

The template custom key value.

Element	l7:Item
----------------	---------

List

Returns a list of custom key values. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/customKeyValues?key=MyKey
```

Returns custom key value with key "MyKey".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/customKeyValues

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
key	string			Key filter

Response

Body

A list of custom key values. If the list is empty then no custom key values were found.

Element	I7:List
----------------	---------

Email Listener

[Top](#)

An email listener will periodically poll an email server for messages to process.

Create

Creates a new email listener

Request

POST 1.0/emailListeners

Body

The email listener to create

Element	I7:EmailListener
Content-Type	application/xml

Response

Body

a reference to the newly email listener

Create Or Update

Creates or Updates an existing email listener. If an email listener with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

```
PUT 1.0/emailListeners/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the email listener to create or update

Body

Email listener to create or update

Element	I7:EmailListener
Content-Type	application/xml

Response

Body

A reference to the newly created or updated email listener.

Delete

Deletes an existing email listener.

Request

```
DELETE 1.0/emailListeners/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the email listener to delete.

Response

No Response Body

Get

Returns an email listener with the given ID.

Request

```
GET 1.0/emailListeners/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the email listener to return

Response

Body

The email listener.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example email listener that can be used as a reference for what email listener objects should look like.

Request

```
GET 1.0/emailListeners/template
```

Response

Body

The template email listener.

Element	l7:Item
----------------	---------

List

Returns a list of email listeners. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/emailListeners?name=MyEmailListener
```

Returns email listener with name "MyEmailListener".

```
/restman/1.0/emailListeners?serverTypes=IMAP
```

Returns email listeners of IMAP type

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/emailListeners

Query Parameters

Param	Type	Default	Since	Description
sort	id name host serverType			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
active	boolean			Active filter
serverType	string			Server type filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of email listener. If the list is empty then no email listeners were found.

Element	I7:List
----------------	---------

Encapsulated Assertion

[Top](#)

Encapsulated Assertion lets you turn any policy fragment into a self-contained "assertion" that accepts input values and sets output values.

Create

Creates a new encapsulated assertion

Request

POST 1.0/encapsulatedAssertions

Body

The encapsulated assertion to create

Element	I7:EncapsulatedAssertion
Content-Type	application/xml

Response

Body

A reference to the newly created encapsulated assertion

Get Dependencies

Returns the list of dependencies for this entity.

Request

GET 1.0/encapsulatedAssertions/{id}/dependencies
--

Query Parameters

Param	Type	Default	Since	Description
level	int	-1	1.0.1	How deep to search for the dependencies. 0 for none, 1 for immediate dependencies, -1 for full depth search

Response

Body

The list of dependencies.

Element	I7:Item
----------------	---------

Create Or Update

Creates or Updates an existing encapsulated assertion. If an encapsulated assertion with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/encapsulatedAssertions/{id}

Path Parameters

Param	Type	Description
id	string	ID of the encapsulated assertion to create or update

Body

Encapsulated assertion to create or update

Element	I7:EncapsulatedAssertion
Content-Type	application/xml

Response

Body

A reference to the newly created or updated encapsulated assertion.

Delete

Deletes an existing encapsulated assertion.

Request

```
DELETE 1.0/encapsulatedAssertions/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the encapsulated assertion to delete.

Response

No Response Body

Get

Returns an encapsulated assertion with the given ID.

Request

```
GET 1.0/encapsulatedAssertions/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the encapsulated assertion to return

Response

Body

The encapsulated assertion.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example encapsulated assertion that can be used as a reference for what encapsulated assertion objects should look like.

Request

```
GET 1.0/encapsulatedAssertions/template
```

Response

Body

The template encapsulated assertion.

Element	l7:Item
----------------	---------

List

Returns a list of encapsulated assertions. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/activeConnectors?name=MyEncass

Returns encapsulated assertion with name "MyEncass".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/encapsulatedAssertions

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
policy.id	string			Policy id filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of encapsulated assertions. If the list is empty then no encapsulated assertions were found.

Element	l7:List
----------------	---------

Firewall Rule

[Top](#)

Since: 1.0.1

A firewall rule is a rule to accept or redirect traffic. This is only available on hardware

installations.

Create

Creates a new firewall rule

Request

POST 1.0/firewallRules

Body

The firewall rule to create

Element	I7:FirewallRule
Content-Type	application/xml

Response

Body

A reference to the newly created firewall rule

Delete

Deletes an existing firewall rule.

Request

DELETE 1.0/firewallRules/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the firewall rule to delete.

Response

No Response Body

Get

Returns a firewall rule with the given ID.

Request

GET 1.0/firewallRules/{id}

Path Parameters

--	--	--

Param	Type	Description
id	string	The ID of the firewall rule to return

Response

Body

The firewall rule.

Element	l7:Item
----------------	---------

Update

Creates or Updates an existing firewall rule. If a firewall rule with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/firewallRules/{id}

Path Parameters

Param	Type	Description
id	string	ID of the firewall rule to create or update

Body

Listen port to create or update

Element	l7:FirewallRule
Content-Type	application/xml

Response

Body

A reference to the newly created or updated firewall rule.

Template

Returns a template, which is an example firewall rule that can be used as a reference for what firewall rule objects should look like.

Request

GET 1.0/firewallRules/template

Response

Body

The template firewall rule.

Element	I7:Item
----------------	---------

List

Returns a list of firewall rules. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/firewallRules?name=MyFirewallRule
```

Returns firewall rule with name "MyFirewallRule".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/firewallRules

Query Parameters

Param	Type	Default	Since	Description
sort	id name enabled ordinal			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
enabled	boolean			Enabled filter
ordinal	string			Ordinal filter

Response

Body

A list of firewall rules. If the list is empty then no firewall rules were found.

Element	I7:List
----------------	---------

Folder

Folders are used to organize the policies, services, and aliases you have on the Gateway.

[Top](#)

Create

Creates a new folder

Request

POST 1.0/folders

Body

The folder to create

Element	l7:Folder
Content-Type	application/xml

Response

Body

A reference to the newly created folder

Get Dependencies

Returns the list of dependencies for this entity.

Request

GET 1.0/folders/{id}/dependencies

Query Parameters

Param	Type	Default	Since	Description
level	int	-1	1.0.1	How deep to search for the dependencies. 0 for none, 1 for immediate dependencies, -1 for full depth search

Response

Body

The list of dependencies.

Element	l7:Item
----------------	---------

Create Or Update

Creates or Updates an existing folder. If a folder with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/folders/{id}

Path Parameters

Param	Type	Description
id	string	ID of the folder to create or update

Body

Folder to create or update

Element	l7:Folder
Content-Type	application/xml

Response

Body

A reference to the newly created or updated folder.

Delete

Deletes an existing folder.

Request

DELETE 1.0/folders/{id}

Path Parameters

Param	Type	Description
id	string	The id of the folder to delete.

Query Parameters

Param	Type	Default	Since	Description
force	boolean	false		If true, deletes folder and its contents

Response

No Response Body

Get

Returns a folder with the given ID.

Request

GET 1.0/folders/{id}

Path Parameters

Param	Type	Description

id	string	The ID of the folder to return
----	--------	--------------------------------

Response

Body

The folder.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example folder that can be used as a reference for what folder objects should look like.

Request

GET 1.0/folders/template

Response

Body

The template folder.

Element	l7:Item
----------------	---------

List

Returns a list of folders. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/folders?name=MyFolder

Returns folder with name "MyFolder".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/folders

Query Parameters

Param	Type	Default	Since	Description
sort	id name parentFolder.id			Key to sort the list by

order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
parentFolder.id	string			The parent folder filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of folders. If the list is empty then no folders were found.

Element	I7:List
----------------	---------

Generic Entity

[Top](#)

A generic entities is used by some modular assertions to represent any entity.

Create

Creates a new generic entity.

Request

POST 1.0/genericEntities

Body

The generic entity to create

Element	I7:GenericEntity
Content-Type	application/xml

Response

Body

A reference to the newly created generic entity

Create Or Update

Creates or Updates an existing generic entity. If a generic entity with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/genericEntities/{id}

Path Parameters

Param	Type	Description
id	string	ID of the generic entity to create or update

Body

Generic entity to create or update

Element	I7:GenericEntity
Content-Type	application/xml

Response

Body

A reference to the newly created or updated generic entity.

Delete

Deletes an existing generic entity.

Request

```
DELETE 1.0/genericEntities/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the generic entity to delete.

Response

No Response Body

Get

Retrieves a generic entity given its ID

Request

```
GET 1.0/genericEntities/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the generic entity to retrieve

Response

Body

The generic entity.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example generic entity that can be used as a reference for what generic entity objects should look like.

Request

GET 1.0/genericEntities/template

Response

Body

The template generic entity.

Element	l7:Item
----------------	---------

List

Returns a list of generic entities. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/activeConnectors?name=MyGenericEntity
```

Returns generic entity with name "MyGenericEntity".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/genericEntities

Query Parameters

Param	Type	Default	Since	Description
sort	id name entityClassName			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter

enabled	boolean			Enabled filter
entityClassName	string			Entity class name filter

Response

Body

A list of generic entities. If the list is empty then no entities were found.

Element	I7:List
----------------	---------

Group

[Top](#)

A group represents a group identity in an identity provider. When no identity provider is specified in the url then the internal identity provider is assumed. Groups can only be retrieved, they can not be created.

Get Group

Returns a group with the given ID.

Request

GET 1.0/groups/{groupID}

Path Parameters

Param	Type	Description
groupID	string	The ID of the group to return

Response

Body

The group.

Element	I7:Item
----------------	---------

Group Template

Returns a template, which is an example group that can be used as a reference for what group objects should look like.

Request

GET 1.0/groups/template

Response

Body

The template group.

Element	l7:Item
----------------	---------

List Groups

Returns a list of groups. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/groups

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter

Response

Body

A list of groups. If the list is empty then no groups were found.

Element	l7:List
----------------	---------

Http Configuration

[Top](#)

HTTP Configuration are used to configure various options to be used by the Gateway for HTTP/HTTPS connections. For example, you can configure the login credentials for an HTTPS host, define a proxy for the host, or specify a private key to be used for authentication.

Create

Creates a new HTTP configuration.

Request

POST 1.0/httpConfigurations

Body

The HTTP configuration to create

Element	l7:HttpConfiguration
Content-Type	application/xml

Response

Body

A reference to the newly created http configuration

Delete

Deletes an existing HTTP configuration.

Request

DELETE 1.0/httpConfigurations/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the HTTP configuration to delete.

Response

No Response Body

Get

Returns an HTTP configuration with the given ID.

Request

GET 1.0/httpConfigurations/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the HTTP configuration to return

Response

Body

The HTTP configuration.

Element	l7:Item
----------------	---------

Update

Creates or Updates an existing HTTP configuration. If an HTTP configuration with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

```
PUT 1.0/httpConfigurations/{id}
```

Path Parameters

Param	Type	Description
id	string	ID of the HTTP configuration to create or update

Body

HTTP configuration to create or update

Element	I7:HttpConfiguration
Content-Type	application/xml

Response

Body

A reference to the newly created or updated HTTP configuration.

Template

Returns a template, which is an example HTTP configuration that can be used as a reference for what HTTP configuration objects should look like.

Request

```
GET 1.0/httpConfigurations/template
```

Response

Body

The template HTTP configuration.

Element	I7:Item
----------------	---------

List

Returns a list of HTTP configurations. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/httpConfigurations?name=MyHTTPConfiguration

Returns HTTP configuration with name "MyHTTPConfiguration".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/httpConfigurations

Query Parameters

Param	Type	Default	Since	Description
sort	id host			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
host	string			Name filter
protocol	HTTP HTTPS			Protocol filter
ntlmHost	string			NtlmHost filter
ntlmDomain	string			NtlmDomain id filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of HTTP configurations. If the list is empty then no HTTP configurations were found.

Element | 7:List

Identity Provider

[Top](#)

An Identity provider is used to store and provide identities.

Create

Creates an identity provider

Request

POST 1.0/identityProviders

Body

The identity provider to create

Element	I7:IdentityProvider
Content-Type	application/xml

Response

Body

A reference to the newly created identity provider

Get Group

Returns a group with the given ID.

Request

GET 1.0/identityProviders/{identityProviderID}/groups/{groupID}

Path Parameters

Param	Type	Description
groupID	string	The ID of the group to return

Response

Body

The group.

Element	I7:Item
----------------	---------

Group Template

Returns a template, which is an example group that can be used as a reference for what group objects should look like.

Request

GET 1.0/identityProviders/{identityProviderID}/groups/template
--

Response

Body

The template group.

Element	I7:Item
----------------	---------

List Groups

Returns a list of groups. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/identityProviders/{identityProviderID}/groups

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter

Response

Body

A list of groups. If the list is empty then no groups were found.

Element	I7:List
----------------	---------

Create Or Update

Creates or Updates an existing identity provider. If an identity provider with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/identityProviders/{identityProviderID}
--

Path Parameters

Param	Type	Description
identityProviderID	string	ID of the identity provider to create or update

Body

Identity provider to create or update

Element	I7:IdentityProvider
Content-Type	application/xml

Response

Body

A reference to the newly created or updated identity provider.

Delete

Deletes an existing identity provider.

Request

```
DELETE 1.0/identityProviders/{identityProviderID}
```

Path Parameters

Param	Type	Description
identityProviderID	string	The ID of the identity provider to delete.

Response

No Response Body

Get

Returns an identity provider with the given ID.

Request

```
GET 1.0/identityProviders/{identityProviderID}
```

Path Parameters

Param	Type	Description
identityProviderID	string	The ID of the identity provider to return

Response

Body

The identity provider.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example identity provider that can be used as a reference for what identity provider objects should look like.

Request

GET 1.0/identityProviders/template

Response

Body

The template identity provider.

Element	7:Item
----------------	--------

List

Returns a list of identity providers. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/identityProviders?name=MyIDProvider

Returns identity provider with name "MyIDProvider".

/restman/1.0/identityProviders?type=LDAP

Returns identity providers of LDAP type

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/identityProviders

Query Parameters

Param	Type	Default	Since	Description
sort	id name type			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
type	LDAP Internal Federated Simple LDAP Policy- Backed			Type filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of entities. If the list is empty then no entities were found.

Element	I7:List
----------------	---------

Create User

Creates a new user. New users can only be created on the internal identity provider.

Request

POST 1.0/identityProviders/{identityProviderID}/users

Body

The user to create.

Element	I7:User
Content-Type	application/xml

Response

Body

A reference to the newly created user

List Users

Returns a list of users. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/identityProviders/{identityProviderID}/users
--

Query Parameters

Param	Type	Default	Since	Description
sort	id login			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
login	string			Login filter

Response

Body

A list of groups. If the list is empty then no groups were found.

Element	I7:List
----------------	---------

Delete User

Deletes an existing user

Request

DELETE 1.0/identityProviders/{identityProviderID}/users/{userID}
--

Path Parameters

Param	Type	Description
userID	string	The ID of the user to delete.

Response

No Response Body

Get User

Returns a user with the given ID.

Request

GET 1.0/identityProviders/{identityProviderID}/users/{userID}

Path Parameters

Param	Type	Description
userID	string	The ID of the user to return

Response

Body

The user.

Element	I7:Item
----------------	---------

Update User

Updates an existing user

Request


```
PUT 1.0/identityProviders/{identityProviderID}/users/{userID}
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user to update

Body

The updated user

Element	I7:User
Content-Type	application/xml

Response

Body

A reference to the newly updated user.

Delete User Certificate

Removes the certificate from the user

Request

```
DELETE 1.0/identityProviders/{identityProviderID}/users/{userID}/certificate
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Response

No Response Body

Get User Certificate

Gets the user's certificate

Request

```
GET 1.0/identityProviders/{identityProviderID}/users/{userID}/certificate
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Response

Body

The certificate

Element	I7:Item
----------------	---------

Set User Certificate

Set this user's certificate

Request

PUT 1.0/identityProviders/{identityProviderID}/users/{userID}/certificate

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Body

The certificate data

Element	I7:CertificateData
Content-Type	application/xml

Response

Body

The certificate set on the user

Element	I7:Item
----------------	---------

Change User Password

Change this user's password

Request

PUT 1.0/identityProviders/{identityProviderID}/users/{userID}/password
--

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Query Parameters

--	--	--	--	--	--

Param	Type	Default	Since	Description
format	string	plain		The format of the password. "plain" or "sha512crypt"

Body

The new password

Element	xsd:string
Content-Type	application/xml

Response

Body

The user that the password was changed for.

Element	l7:Item
----------------	---------

Template User

Returns a template, which is an example user that can be used as a reference for what user objects should look like.

Request

```
GET 1.0/identityProviders/{identityProviderID}/users/template
```

Response

Body

The template user.

Element	l7:Item
----------------	---------

Interface Tag

[Top](#)

Interfaces are used to specify IP addresses that can be monitored by a listen port. Defining an interface gives you greater control over the IP addresses that will be monitored.

Create

Creates a new interface. In order to create a new interface tag you must have read and write access to cluster properties.

Request

POST 1.0/interfaceTags

Body

The interface to create

Element	I7:InterfaceTag
Content-Type	application/xml

Response

Body

A reference to the newly created interface

Delete

Deletes an existing interface tag.

Request

DELETE 1.0/interfaceTags/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the interface tag to delete.

Response

No Response Body

Get

Retrieves an interface tag given it's ID. In order to get an interface tag you must have read access to cluster properties.

Request

GET 1.0/interfaceTags/{id}

Path Parameters

Param	Type	Description
id	string	The identity of the interface tag to select

Response

Body

The interface tag.

Element	l7:Item
----------------	---------

Update

Updates an existing interface tag. You cannot change the name of an interface, you can only update its address patterns. In order to update an interface tag you must have read and write access to cluster properties.

Request

PUT 1.0/interfaceTags/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the interface tag to update

Body

The updated interface tag

Element	l7:InterfaceTag
Content-Type	application/xml

Response

Body

a reference to the newly updated interface tag.

Template

Returns a template, which is an example interface tag that can be used as a reference for what interface tag objects should look like.

Request

GET 1.0/interfaceTags/template

Response

Body

The template interface tag.

Element	l7:Item
----------------	---------

List

Returns a list of interface tags. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/interfaceTags?name=MyInterface
```

Returns interface with name = "MyInterface"

```
/restman/interfaceTags?name=MyInterface&name=MyInterfaceProd
```

Returns interfaces with name either "MyInterface" or "MyInterfaceProd"

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/interfaceTags

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter. This will return interfaces with the specified names.

Response

Body

A list of interfaces. If the list is empty then no entities were found.

Element	7:List
----------------	--------

JDBC Connection

[Top](#)

JDBC connections allow the Gateway to query external databases and then use the query results during policy consumption.

Create

Creates a new JDBC connection

Request

POST 1.0/jdbcConnections

Body

The JDBC connection to create

Element	I7:JDBCCConnection
Content-Type	application/xml

Response

Body

A reference to the newly created JDBC connection

Create Or Update

Creates or Updates an existing JDBC connection. If a JDBC connection with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/jdbcConnections/{id}

Path Parameters

Param	Type	Description
id	string	ID of the JDBC connection to create or update

Body

JDBC connection to create or update

Element	I7:JDBCCConnection
Content-Type	application/xml

Response

Body

A reference to the newly created or updated JDBC connection.

Delete

Deletes an existing JDBC connection.

Request

```
DELETE 1.0/jdbcConnections/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the JDBC connection to delete.

Response

No Response Body

Get

Returns a JDBC connection with the given ID.

Request

```
GET 1.0/jdbcConnections/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the JDBC connection to return

Response

Body

The JDBC connection.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example JDBC connection that can be used as a reference for what JDBC connection objects should look like.

Request

```
GET 1.0/jdbcConnections/template
```

Response

Body

The template JDBC connection.

Element	l7:Item
----------------	---------

List

Returns a list of JDBC connections. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/jdbcConnections?name=MyJDBCConnection
```

Returns JDBC connection with name "MyJDBCConnection".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/jdbcConnections

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
enabled	boolean			Enabled filter
jdbcUrl	string			JDBC URL filter
driverClass	string			Driver class name filter
userName	string			User name filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of JDBC connections. If the list is empty then no JDBC connections were found.

Element	7:List
----------------	--------

JMS Destination

[Top](#)

A JMS destination is used to configure a connection to a JMS service. This is used for both inbound and outbound configurations.

Create

Creates a new JMS destination

Request

POST 1.0/jmsDestinations

Body

The JMS destination to create

Element	I7:JMSDestination
Content-Type	application/xml

Response

Body

A reference to the newly created JMS destination

Create Or Update

Creates or Updates an existing JMS destination. If an JMS destination with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/jmsDestinations/{id}

Path Parameters

Param	Type	Description
id	string	ID of the JMS destination to create or update

Body

JMS destination to create or update

Element	I7:JMSDestination
Content-Type	application/xml

Response

Body

A reference to the newly created or updated JMS destination.

Delete

Deletes an existing JMS destination.

Request

DELETE 1.0/jmsDestinations/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the JMS destination to delete.

Response

No Response Body

Get

Returns an JMS destination with the given ID.

Request

GET 1.0/jmsDestinations/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the JMS destination to return

Response

Body

The JMS destination.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example JMS destination that can be used as a reference for what JMS destination objects should look like.

Request

GET 1.0/jmsDestinations/template

Response

Body

The template JMS destination.

Element	l7:Item
----------------	---------

List

Returns a list of JMS destinations. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/jmsDestinations?name=MyJMSDestination
```

Returns JMS destination with name "MyJMSDestination".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/jmsDestinations

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
enabled	boolean			Enabled filter
inbound	boolean			Inbound filter
template	boolean			Template filter
destination	string			Destination filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of JMS destinations. If the list is empty then no JMS destinations were found.

Element	I7:List
----------------	---------

Listen Port

[Top](#)

A listen port is a TCP port that "listens" for incoming messages that are then passed to the Gateway message processor.

Create

Creates a new listen port

Request

POST 1.0/listenPorts

Body

The listen port to create

Element	I7:ListenPort
Content-Type	application/xml

Response

Body

A reference to the newly created listen port

List

Returns a list of listen ports. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/listenPorts?name=MyListenPort

Returns listen port with name "MyListenPort".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/listenPorts

Query Parameters

Param	Type	Default	Since	Description
sort	id name enabled protocol port			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
enabled	boolean			Enabled filter
protocol	string			Protocol filter
port	string			Port filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of listen ports. If the list is empty then no listen ports were found.

Element	I7:List
----------------	---------

Delete

Deletes an existing listen port.

Request

DELETE 1.0/listenPorts/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the listen port to delete.

Response

No Response Body

Get

Returns a listen port with the given ID.

Request

GET 1.0/listenPorts/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the listen port to return

Response

Body

The listen port.

Element	I7:Item
----------------	---------

Update

Creates or Updates an existing listen port. If a listen port with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/listenPorts/{id}

Path Parameters

Param	Type	Description
id	string	ID of the listen port to create or update

Body

Listen port to create or update

Element	I7:ListenPort
Content-Type	application/xml

Response

Body

A reference to the newly created or updated listen port.

Template

Returns a template, which is an example listen port that can be used as a reference for what listen port objects should look like.

Request

GET 1.0/listenPorts/template

Response

Body

The template listen port.

Element	I7:Item
----------------	---------

Policy

[Top](#)

This resource is used to manage policies. These can be internal, global or other special purpose policies.

Create

Creates a new policy

Request

POST 1.0/policies

Query Parameters

Param	Type	Default	Since	Description
versionComment	string			The comment to add to the policy version when creating the policy

Body

The policy to create

Element	I7:Policy
Content-Type	application/xml

Response

Body

A reference to the newly created policy

Get Dependencies

Returns the list of dependencies for this entity.

Request

GET 1.0/policies/{id}/dependencies

Query Parameters

Param	Type	Default	Since	Description
level	int	-1	1.0.1	How deep to search for the dependencies. 0 for none, 1 for immediate dependencies, -1 for full depth search

Response

Body

The list of dependencies.

Element	I7:Item
----------------	---------

List

Returns a list of policies. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/policies?name=MyPolicy

Returns policy with name "MyPolicy".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/policies

Query Parameters

Param	Type	Default	Since	Description
sort	id name parentFolder.id			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
guid	string			Guid filter
type	Include Internal Global			Type filter
soap	boolean			Soap filter
parentFolder.id	string			Parent folder ID filter.
securityZone.id	string			Security zone ID filter

Response

Body

A list of policies. If the list is empty then no policies were found.

Element	I7:List
----------------	---------

Delete

Deletes an existing policy.

Request

DELETE 1.0/policies/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the policy to delete.

Response

No Response Body

Get

Returns a policy with the given ID.

Request

GET 1.0/policies/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the policy to return

Response

Body

The policy.

Element	I7:Item
----------------	---------

Update Or Create

Creates or Updates an existing policy. If a policy with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/policies/{id}

Path Parameters

Param	Type	Description
id	string	ID of the policy to create or update

Query Parameters

Param	Type	Default	Since	Description
active	boolean	true		Should the policy be activated after the update.
versionComment	string			The comment to add to the policy version when updating the policy

Body

Policy to create or update

Element	I7:Policy
Content-Type	application/xml

Response

Body

A reference to the newly created or updated policy.

Template

Returns a template, which is an example policy that can be used as a reference for what policy objects should look like.

Request

GET 1.0/policies/template

Response

Body

The template policy.

Element	l7:Item
----------------	---------

List Versions

Returns a list of policy versions. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/policies/{id}/versions

Query Parameters

Param	Type	Default	Since	Description
sort	id version			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
id	string			Id filter
active	boolean			Active filter
comment	string			Comment filter

Response

Body

A list of policy versions. If the list is empty then no policy versions were found.

Element	l7:List
Content-Type	application/xml

Get Version

Retrieve a policy version by the version number

Request

```
GET 1.0/policies/{id}/versions/{versionNumber}
```

Path Parameters

Param	Type	Description
versionNumber	long	Version of the policy to return.

Response

Body

The policy version.

Element	l7:Item
----------------	---------

Activate Version

Activates the specified policy version

Request

```
POST 1.0/policies/{id}/versions/{versionNumber}/activate
```

Path Parameters

Param	Type	Description
versionNumber	long	Version of the policy version to set active.

Response

No Response Body

Set Version Comment

Sets the comment on a specific policy version.

Request

PUT 1.0/policies/{id}/versions/{versionNumber}/comment

Path Parameters

Param	Type	Description
versionNumber	long	Version of the policy version to set the comment on.

Body

Comment to set on the policy version. This will override any existing comment.

Element	xsd:string
Content-Type	application/xml

Response

Body

A reference to the updated policy version

Element	l7:Item
----------------	---------

Get Active Version

Returns the active policy version.

Request

GET 1.0/policies/{id}/versions/active

Response

Body

The active policy version.

Element	l7:Item
----------------	---------

Set Active Version Comment

Sets a comment on the active policy version

Request

PUT 1.0/policies/{id}/versions/active/comment

Body

Comment to set on the active policy version

Element	xsd:string
Content-Type	application/xml

Response

Body

The updated active policy version

Element	l7:Item
----------------	---------

Policy Alias

[Top](#)

A policy alias allows a policy to appear in more than one folder in the Services and Policies list. The policy alias is a linked copy of the original policy.

Create

Creates a new policy alias

Request

POST 1.0/policyAliases

Body

The policy alias to create

Element	l7:PolicyAlias
Content-Type	application/xml

Response

Body

A reference to the newly created policy alias

List

Returns a list of policy aliases. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/policyAliases?policy.id=26df9b0abc4dd6780fd9da5929cde13e
```

Returns policy aliases for policy with ID "26df9b0abc4dd6780fd9da5929cde13e".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/policyAliases

Query Parameters

Param	Type	Default	Since	Description
sort	id policy.id folder.id			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
policy.id	string			Policy id filter
folder.id	string			Folder id filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of policy aliases. If the list is empty then no policy aliases were found.

Element	I7:List
----------------	---------

Delete

Deletes an existing policy alias.

Request

DELETE 1.0/policyAliases/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the policy alias to delete.

Response

No Response Body

Get

Returns a policy alias with the given ID.

Request

GET 1.0/policyAliases/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the policy alias to return

Response

Body

The policy alias.

Element	l7:Item
----------------	---------

Update Or Create

Creates or Updates an existing policy alias. If a policy alias with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/policyAliases/{id}

Path Parameters

Param	Type	Description
id	string	ID of the policy alias to create or update

Body

Policy alias to create or update

Element	l7:PolicyAlias
Content-Type	application/xml

Response

Body

A reference to the newly created or updated policy alias.

Template

Returns a template, which is an example policy alias that can be used as a reference for what policy alias objects should look like.

Request

GET 1.0/policyAliases/template

Response

Body

The template policy alias.

Element	l7:Item
----------------	---------

Policy Bundle Installer

[Top](#)

Since: 1.0.1

This resource exposes the Policy Bundle Installer for no-GUI use cases (e.g. support auto provisioning scripting to install OAuth).

Execute

Executes a Policy Bundle Installer action. Choose the installer instance by setting HTTP parameter `installer_name` (e.g. set to `OAuthInstaller`).

The following actions are supported: list components, restman get migration bundle XML, wsman dry run install, wsman install and custom. For restman bundle, get migration bundle. For wsman bundle, execute dry run and install.

Provides actions through the following HTTP parameters:

List components

List component id(s) available for this installer bundle. This is the default if no action parameter is provided.

Input HTTP parameter: set `action` to `list`

Output: List of component id(s), ";" separated. For example:

1c2a2874-df8d-4e1d-b8b0-099b576407e1;ba525763-6e55-4748-9376-76055247c8b1;f69c7d15-4999-4761-ab

Restman get migration bundle

Get the restman migration bundle XML for the given component(s).

Input HTTP parameters:

- Set `action` to `restman_get`
- Set `component_ids` to a ";" separated list of component ids. Or set to `all` to specify all available installer components. This is equivalent to all component id(s) from the `list` action.
- Optionally set `version_modifier` - optional version modifier
- Optionally set `folder_goid` - optional install folder (if not set, defaults to root folder)

Output: The restman migration bundle XML for each component. For example:

```

<l7:Bundle xmlns:l7="http://ns.17tech.com/2010/04/gateway-management">
  <l7:References>
    <l7:Item>
      ...
    </l7:Mapping>
  </l7:Mappings>
</l7:Bundle>

```

Wsman dry run install

Execute wsman dry run install for given component(s).

Input HTTP parameters:

- Set action to `wsman_dry_run`
- Set `component_ids` to a ";" separated list of component ids. Or set to `all` to specify all available installer components. This is equivalent to all component id(s) from the `list` action.
- Optionally set `version_modifier` - optional version modifier
- Optionally map JDBC connection name to new name
 - Set `jdbc_connection.<component_id>.name` with existing name in bundle (e.g. OAuth)
 - Set `jdbc_connection.<component_id>.new_name` with a new desired name (e.g. OAuth Dev)

Output: The conflicts for each component. For example:

```

ComponentId: 1c2a2874-df8d-4e1d-b8b0-099b576407e1
ServiceConflict: /auth/oauth/v1/token;/auth/oauth/v1/authorize/website;/oauth/v1/client;/prote
PolicyConflict: OAuth 1.0 Context Variables;Require OAuth 1.0 Token;getClientSignature;Authent
CertificateConflict:
JdbcConnectionsThatDontExist:
MissingAssertions:
EncapsulatedAssertionConflict:
...

```

It's recommended to **not continue** with installation if any conflicts are detected. Continuing the installation with detected conflicts may result in partial installation of components. Components with detected conflicts will **not** be installed. As a result, references to conflicted components may incorrectly reference the previous version of the component already on the Gateway (if one exists).

Wsman install

Execute wsman install for given component(s).

Input HTTP parameters:

- Set action to `wsman_install`
- Set `component_ids` to a ";" separated list of component ids. Or set to `all` to specify

all available installer components. This is equivalent to all component id(s) from the `list` action.

- Optionally set `version_modifier` - optional version modifier
- Optionally set `folder_goid` - optional install folder (if not set, defaults to root folder)
- Optionally map JDBC connection name to new name
 - Set `jdbc_connection.<component_id>.name` with existing name in bundle (e.g. OAuth)
 - Set `jdbc_connection.<component_id>.new_name` with a new desired name (e.g. OAuth Dev)

Output: List of installed component id(s), ";" separated. For example:

1c2a2874-df8d-4e1d-b8b0-099b576407e1;ba525763-6e55-4748-9376-76055247c8b1;f69c7d15-4999-4761-ab

Custom

An installer can implement custom action, custom wsman dry run and install logic. For example the OAuth installer implements its own custom action to get a database schema. It also executes custom logic to integrate API Portal for wsman dry run and install.

Input: Set HTTP parameter `action` to `custom` in order to choose a custom action. **And** set HTTP parameter(s) required for the custom installer.

Request

GET 1.0/policyBundleInstallers

Query Parameters

Param	Type	Default	Since	Description
installer_name	string			name of installer instance to use (e.g. OAuthInstaller)

Response

Body

The stream output from the installer.

Content-Type	text/xml
---------------------	----------

Private Key

This resource is used to manage private keys.

[Top](#)

List

Returns a list of private keys. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/privateKeys?alias=mykey
```

Returns the private key with alias "mykey"

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/privateKeys

Query Parameters

Param	Type	Default	Since	Description
sort	id			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
alias	string			Alias filter

Response

Body

A list of private keys. If the list is empty then no private keys were found.

Element	7:List
----------------	--------

Create Resource

Creates a new private key given a PrivateKeyCreationContext. The PrivateKeyCreationContext is used to specify how the private key should be created. The PrivateKeyCreationContext dn is the domain name to create the key with. The properties are optional but you may specify:

Key	Type	Description
ecName	String	This the Elliptic Curve key type to use. If it is not specified an RSA key type is used.
rsaKeySize	Integer	This is the rsa key size to use. This is only applicable if an ecName is not specified. Defaults to 2048
daysUntilExpiry	Integer	Specify the number of days until the key expires. Defaults to 5 years.
caCapable	Boolean	Specify if the certificate should be CA capable.

		Defaults to false
signatureHashAlgorithm	String	The algorithm used for the signature hash.

Example request:

```
<l7:PrivateKeyCreationContext xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
  <l7:Dn>CN=srcAlias</l7:Dn>
  <l7:Properties>
    <l7:Property key="signatureHashAlgorithm">
      <l7:StringValue>SHA384</l7:StringValue>
    </l7:Property>
    <l7:Property key="rsaKeySize">
      <l7:IntegerValue>516</l7:IntegerValue>
    </l7:Property>
    <l7:Property key="ecName">
      <l7:StringValue>secp384r1</l7:StringValue>
    </l7:Property>
    <l7:Property key="daysUntilExpiry">
      <l7:IntegerValue>2</l7:IntegerValue>
    </l7:Property>
    <l7:Property key="caCapable">
      <l7:BooleanValue>true</l7:BooleanValue>
    </l7:Property>
  </l7:Properties>
</l7:PrivateKeyCreationContext>
```

This responds with a reference to the newly created private key.

Request

POST 1.0/privateKeys/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the private key to create in the form of [keystore ID]:[alias]

Body

This specifies how to create the private key.

Element	l7:PrivateKeyCreationContext
Content-Type	application/xml

Response

Body

A reference to the newly created private key

Delete

Deletes an existing private key.

Request

```
DELETE 1.0/privateKeys/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the private key to delete.

Response

No Response Body

Get

Retrieve a private key by its ID. The ID is a combination of the keystoreId and the key alias separated by a ':'. For example
0000000000000000000000000000000002:mykey

Request

```
GET 1.0/privateKeys/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the private key to retrieve

Response

Body

The private key.

Element	l7:Item
----------------	---------

Update

Updates an existing private key. This api call can be used to replace a private key's certificate chain or change its security zone.

Request

```
PUT 1.0/privateKeys/{id}
```

Path Parameters

Param	Type	Description

id	string	The ID of the private key to update
----	--------	-------------------------------------

Body

The updated private key.

Element	I7:PrivateKey
Content-Type	application/xml

Response

Body

A reference to the newly updated private key.

Export Private Key

Export a private key.

Request

PUT 1.0/privateKeys/{id}/export

Path Parameters

Param	Type	Description
id	string	The ID of the key to export

Body

The export key context. This contains a password to secure the exported key with. The password must always be specified but can be the empty string to specify no password. This can also specify the alias to export the private key with. If no alias is specified the key will be exported with its current alias.

Element	I7:PrivateKeyExportContext
Content-Type	application/xml

Response

Body

The exported private key

Element	I7:Item
----------------	---------

Generate Csr

Generate a certificate signing request for this private key.

Request

GET 1.0/privateKeys/{id}/generateCSR

Path Parameters

Param	Type	Description
id	string	The ID of the key to generate the CSR from

Query Parameters

Param	Type	Default	Since	Description
csrSubjectDN	string			The CSR subject dn to use. It defaults to the key's subject dn if none is specified.
signatureHash	SHA1 SHA256 SHA384 SHA512			The signature hash to use. Defaults to 'Automatic'

Response

Body

The CSR data.

Element	I7:Item
----------------	---------

Import Private Key

Import a private key.

Request

POST 1.0/privateKeys/{id}/import

Path Parameters

Param	Type	Description
id	string	The ID to import the key into

Body

The private key to import. This contains the key data. It contains the password that the key is secured with. An Alias can also be given to specify which alias to use if the key given contains more then one alias. By default the first alias in the given key is used.

Element	I7:PrivateKeyImportContext
Content-Type	application/xml

Response

Body

A reference to the newly imported private key.

Element	l7:Item
----------------	---------

Sign Cert

Signs a csr pem file with the specified key.

Request

PUT 1.0/privateKeys/{id}/signCert

Path Parameters

Param	Type	Description
id	string	The ID of the key to sign the certificate with

Query Parameters

Param	Type	Default	Since	Description
subjectDN	string			The subject DN to set on the signed certificate
expiryAge	int	730		The expiry age of the certificate
signatureHash	Automatic SHA1 SHA256 SHA384 SHA512	Automatic		The signature hash to use. Defaults to 'Automatic'

Body

The certificate csr to sign

Element	xsd:string
Content-Type	application/xml

Body

Content-Type	application/x-pem-file
---------------------	------------------------

Response

Body

The signed certificate.

Element	l7:Item
----------------	---------

Mark Special Purpose

Mark a private key for a special special purpose

Request

```
PUT 1.0/privateKeys/{id}/specialPurpose
```

Path Parameters

Param	Type	Description
id	string	The ID of the key to mark for special purpose

Query Parameters

Param	Type	Default	Since	Description
purpose	SSL CA AUDIT_VIEWER AUDIT_SIGNING			The special purpose to mark the key with. Can specify more then one special purposes.

Response

Body

A reference to the newly updated private key.

Element	7:Item
----------------	--------

Template

Returns a template, which is an example private key that can be used as a reference for what private key objects should look like.

Request

```
GET 1.0/privateKeys/template
```

Response

Body

The template private key.

Element	7:Item
----------------	--------

Template Private Key Creation Context

Returns a template, which is an example private key creation context that can be used as a reference for what private key creation context objects should look like.

Request

GET 1.0/privateKeys/template/privatekeycreationcontext
--

Response

Body

The template private key creation context.

Element	7:Item
----------------	--------

Template Private Key Export Context

Returns a template, which is an example private key export context that can be used as a reference for what private key export context objects should look like.

Request

GET 1.0/privateKeys/template/privatekeyexportcontext
--

Response

Body

The template private key export context.

Element	7:Item
----------------	--------

Template Private Key Import Context

Returns a template, which is an example private key import context that can be used as a reference for what private key import context objects should look like.

Request

GET 1.0/privateKeys/template/privatekeyimportcontext
--

Response

Body

The template private key import context.

Element	7:Item
----------------	--------

Published Service

This resource is used to manage services.

[Top](#)

Create

Creates a new service

Request

POST 1.0/services

Query Parameters

Param	Type	Default	Since	Description
versionComment	string			The comment to add to the policy version when creating the service

Body

The service to create

Element	I7:Service
Content-Type	application/xml

Response

Body

A reference to the newly created service

Get Dependencies

Returns the list of dependencies for this entity.

Request

GET 1.0/services/{id}/dependencies

Query Parameters

Param	Type	Default	Since	Description
level	int	-1	1.0.1	How deep to search for the dependencies. 0 for none, 1 for immediate dependencies, -1 for full depth search

Response

Body

The list of dependencies.

Element	I7:Item
----------------	---------

List

Returns a list of servicees. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/services?name=MyService
```

Returns service with name "MyService".

If a parameter is not a valid search value a bad request error will be returned.

Request

```
GET 1.0/services
```

Query Parameters

Param	Type	Default	Since	Description
sort	id name parentFolder.id			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
guid	string			Guid filter
enabled	boolean			Enabled filter
soap	boolean			Soap filter
parentFolder.id	string			Parent folder ID filter.
securityZone.id	string			Security zone ID filter

Response

Body

A list of services. If the list is empty then no services were found.

Element	7:List
----------------	--------

List Versions

Returns a list of policy versions. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

If a parameter is not a valid search value a bad request error will be returned.

Request

```
GET 1.0/services/{id}/versions
```

Query Parameters

Param	Type	Default	Since	Description
sort	id version			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
id	string			Id filter
active	boolean			Active filter
comment	string			Comment filter

Response

Body

A list of policy versions. If the list is empty then no policy versions were found.

Element	I7:List
Content-Type	application/xml

Get Version

Retrieve a policy version by the version number

Request

```
GET 1.0/services/{id}/versions/{versionNumber}
```

Path Parameters

Param	Type	Description
versionNumber	long	Version of the policy to return.

Response

Body

The policy version.

Element	I7:Item
----------------	---------

Activate Version

Activates the specified policy version

Request

```
POST 1.0/services/{id}/versions/{versionNumber}/activate
```

Path Parameters

Param	Type	Description
versionNumber	long	Version of the policy version to set active.

Response

No Response Body

Set Version Comment

Sets the comment on a specific policy version.

Request

```
PUT 1.0/services/{id}/versions/{versionNumber}/comment
```

Path Parameters

Param	Type	Description
versionNumber	long	Version of the policy version to set the comment on.

Body

Comment to set on the policy version. This will override any existing comment.

Element	xsd:string
Content-Type	application/xml

Response

Body

A reference to the updated policy version

Element	l7:Item
----------------	---------

Get Active Version

Returns the active policy version.

Request

```
GET 1.0/services/{id}/versions/active
```

Response

Body

The active policy version.

--	--

Element	l7:Item
----------------	---------

Set Active Version Comment

Sets a comment on the active policy version

Request

PUT 1.0/services/{id}/versions/active/comment

Body

Comment to set on the active policy version

Element	xsd:string
Content-Type	application/xml

Response

Body

The updated active policy version

Element	l7:Item
----------------	---------

Delete

Deletes an existing service.

Request

DELETE 1.0/services/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the service to delete.

Response

No Response Body

Get

Returns a service with the given ID.

Request

GET 1.0/services/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the service to return

Response

Body

The service.

Element	l7:Item
----------------	---------

Update Or Create

Creates or Updates an existing service. If a service with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/services/{id}

Path Parameters

Param	Type	Description
id	string	ID of the service to create or update

Query Parameters

Param	Type	Default	Since	Description
active	boolean	true		Should the service be activated after the update.
versionComment	string			The comment to add to the policy version when updating the service

Body

Service to create or update

Element	l7:Service
Content-Type	application/xml

Response

Body

A reference to the newly created or updated service.

Template

Returns a template, which is an example service that can be used as a reference for what service objects should look like.

Request

GET 1.0/services/template

Response

Body

The template service.

Element	l7:Item
----------------	---------

Resource Document

[Top](#)

Resource documents are documents schema documents. They are either a dtd or an xml schema.

Create

Creates a new resource document

Request

POST 1.0/resources

Body

The resource document to create

Element	l7:ResourceDocument
Content-Type	application/xml

Response

Body

a reference to the newly created resource document

List

Returns a list of resource documents. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/resources?uri=documentURI

Returns resource document with uri "documentURI".

/restman/1.0/resources?type=xmlschema

Returns resource documents of xmlschema type

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/resources

Query Parameters

Param	Type	Default	Since	Description
sort	id uri			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
uri	string			Uri filter
description	string			Description filter
type	dtd xmlschema			Type filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of resource documents. If the list is empty then no resource documents were found.

Element	l7:List
----------------	---------

Delete

Deletes an existing resource document.

Request

DELETE 1.0/resources/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the resource document to delete.

Response

No Response Body

Get

Returns a resource document with the given ID.

Request

GET 1.0/resources/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the resource document to return

Response

Body

The resource document.

Element	I7:Item
----------------	---------

Update

Creates or Updates an existing resource document. If a resource document with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/resources/{id}

Path Parameters

Param	Type	Description
id	string	ID of the resource document to create or update

Body

Resource document to create or update

Element	I7:ResourceDocument
Content-Type	application/xml

Response

Body

A reference to the newly created or updated resource document.

Template

Returns a template, which is an example resource document that can be used as a reference for what resource document objects should look like.

Request

GET 1.0/resources/template

Response

Body

The template resource document.

Element	l7:Item
----------------	---------

Revocation Checking Policy

[Top](#)

A revocation checking policy defines the strategies used by the Gateway to determine whether a certificate has been revoked.

Create

Creates a new revocation checking policy

Request

POST 1.0/revocationCheckingPolicies

Body

The revocation checking policy to create

Element	l7:RevocationCheckingPolicy
Content-Type	application/xml

Response

Body

A reference to the newly created revocation checking policy

List

Returns a list of revocation checking policies. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/revocationCheckingPolicies?name=MyRevocationCheckPolicy

Returns revocation checking policy with name "MyRevocationCheckPolicy".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/revocationCheckingPolicies

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of revocation checking policies. If the list is empty then no revocation checking policies were found.

Element	I7:List
----------------	---------

Create Or Update

Creates or Updates an existing revocation checking policy. If an revocation checking policy with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/revocationCheckingPolicies/{id}

Path Parameters

Param	Type	Description
id	string	ID of the revocation checking policy to create or update

Body

Revocation checking policy to create or update

Element	I7:RevocationCheckingPolicy
Content-Type	application/xml

Response

Body

A reference to the newly created or updated revocation checking policy.

Delete

Deletes an existing revocation checking policy.

Request

```
DELETE 1.0/revocationCheckingPolicies/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the revocation checking policy to delete.

Response

No Response Body

Get

Returns a revocation checking policy with the given ID.

Request

```
GET 1.0/revocationCheckingPolicies/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the revocation checking policy to return

Response

Body

The revocation checking policy.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example revocation checking policy that can be used as a reference for what revocation checking policy objects should look like.

Request

GET 1.0/revocationCheckingPolicies/template

Response

Body

The template revocation checking policy.

Element	l7:Item
----------------	---------

Role

[Top](#)

A role defines a set of permissions that a user or group is allowed to perform. A user must be assigned to at least one of these roles in order to connect to the Gateway and perform administrative tasks in the Policy Manager or through the API.

Create

Creates a new role

Request

POST 1.0/roles

Body

The role to create

Element	l7:Role
Content-Type	application/xml

Response

Body

A reference to the newly created role

List

Returns a list of roles. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/roles?name=MyRole

Returns role with name "MyRole".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/roles

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
userCreated	boolean			User created filter

Response

Body

A list of roles. If the list is empty then no roles were found.

Element	I7:List
----------------	---------

Create Or Update

Creates or Updates an existing role. If a role with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/roles/{id}

Path Parameters

Param	Type	Description
id	string	ID of the role to create or update

Body

Role to create or update

Element	I7:Role
Content-Type	application/xml

Response

Body

A reference to the newly created or updated role.

Delete

Deletes an existing role.

Request

```
DELETE 1.0/roles/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the role to delete.

Response

No Response Body

Get

Returns a role with the given ID.

Request

```
GET 1.0/roles/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the role to return

Response

Body

The role

Element	l7:Item
----------------	---------

Add Assignment

Adds role assignments to the role.

Request

```
PUT 1.0/roles/{id}/assignments
```

Body

The role assignments to add.

Element	I7:AddAssignmentsContext
Content-Type	application/xml

Response

No Response Body

Delete Assignment

Removes role assignments

Request

```
DELETE 1.0/roles/{id}/assignments
```

Query Parameters

Param	Type	Default	Since	Description
id	string			The assignments to remove

Response

No Response Body

Add Assignments Context Template

Returns a template, which is an example add assignments context that can be used as a reference for what add assignments context objects should look like.

Request

```
GET 1.0/roles/{id}/assignments/template/addassignments
```

Response

Body

The template add assignments context.

Element	I7:Item
----------------	---------

Template

Returns a template, which is an example role that can be used as a reference for what role objects should look like.

Request

```
GET 1.0/roles/template
```

Response

Body

The template role.

Element	l7:Item
----------------	---------

Sample Message

[Top](#)

Since: 1.0.1

A sample message is a message to help configure assertions.

Create

Creates a new sample message

Request

POST 1.0/sampleMessages

Body

The sample message to create

Element	l7:SampleMessage
Content-Type	application/xml

Response

Body

A reference to the newly created sample message

List

Returns a list of sample messages. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/sampleMessages?name=MySampleMessage
```

Returns sample message with name "MySampleMessage".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/sampleMessages

Query Parameters

Param	Type	Default	Since	Description
sort	id name operationName service.id			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
operationName	string			Operation filter
service.id	string			Service ID filter

Response

Body

A list of sample messages. If the list is empty then no sample messages were found.

Element	7:List
----------------	--------

Delete

Deletes an existing sample message.

Request

DELETE 1.0/sampleMessages/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the sample message to delete.

Response

No Response Body

Get

Returns a sample message with the given ID.

Request

GET 1.0/sampleMessages/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the sample message to return

Response

Body

The sample message.

Element	I7:Item
----------------	---------

Update

Creates or Updates an existing sample message. If a sample message with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/sampleMessages/{id}

Path Parameters

Param	Type	Description
id	string	ID of the sample message to create or update

Body

Sample message to create or update

Element	I7:SampleMessage
Content-Type	application/xml

Response

Body

A reference to the newly created or updated sample message.

Template

Returns a template, which is an example sample message that can be used as a reference for what sample message objects should look like.

Request

GET 1.0/sampleMessages/template

Response

Body

The template sample message.

Element	l7:Item
----------------	---------

Secure Password

[Top](#)

Secure passwords are used to securely store passwords and plain text PEM private keys in the Gateway database.

Create

Creates a new secure password

Request

POST 1.0/passwords

Body

The secure password to create

Element	l7:StoredPassword
Content-Type	application/xml

Response

Body

A reference to the newly created secure password

List

Returns a list of secure passwords. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/passwords?name=MyPassword
```

Returns secure password with name "MyPassword".

```
/restman/1.0/passwords?type=Password&name=MyPassword&name=MyOtherPassword
```

Returns secure password of Password type with name either "MyPassword" or "MyOtherPassword"

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/passwords

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
type	Password PEM Private Key			Type filter

Response

Body

A list of secure passwords. If the list is empty then no secure passwords were found.

Element	I7:List
----------------	---------

Create Or Update

Creates or Updates an existing secure password. If a secure password with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/passwords/{id}

Path Parameters

Param	Type	Description
id	string	ID of the secure password to create or update

Body

Secure password to create or update

Element	I7:StoredPassword
Content-Type	application/xml

Response

Body

A reference to the newly created or updated secure password.

Delete

Deletes an existing secure password.

Request

```
DELETE 1.0/passwords/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the secure password to delete.

Response

No Response Body

Get

Returns a secure password with the given ID.

Request

```
GET 1.0/passwords/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the secure password to return

Response

Body

The secure password.

Element	l7:Item
----------------	---------

Template

Returns a template, which is an example secure password that can be used as a reference for what secure password objects should look like.

Request

GET 1.0/passwords/template

Response

Body

The template secure password.

Element	l7:Item
----------------	---------

Security Zone

[Top](#)

Security zones are used to partition the Gateway into portions that can then be managed by other users. A security zone is a collection of related entities (for example: services, policies, folders, trusted certificates).

Create

Creates a new security zone

Request

POST 1.0/securityZones

Body

The security zone to create

Element	l7:SecurityZone
Content-Type	application/xml

Response

Body

A reference to the newly created security zone

List

Returns a list of security zones. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

/restman/1.0/securityZones?name=MySecurityZone

Returns security zone with name "MySecurityZone".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/securityZones

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter

Response

Body

A list of security zones. If the list is empty then no security zones were found.

Element	I7:List
----------------	---------

Create Or Update

Creates or Updates an existing security zone. If an security zone with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/securityZones/{id}

Path Parameters

Param	Type	Description
id	string	ID of the security zone to create or update

Body

Security zone to create or update

Element	I7:SecurityZone
Content-Type	application/xml

Response

Body

A reference to the newly created or updated security zone.

Delete

Deletes an existing security zone.

Request

```
DELETE 1.0/securityZones/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the security zone to delete.

Response

No Response Body

Get

Returns a security zone with the given ID.

Request

```
GET 1.0/securityZones/{id}
```

Path Parameters

Param	Type	Description
id	string	The ID of the security zone to return

Response

Body

The security zone

Element	7:Item
----------------	--------

Template

Returns a template, which is an example security zone that can be used as a reference for what security zone objects should look like.

Request

```
GET 1.0/securityZones/template
```

Response

Body

The template security zone

Element

l7:Item

Service Alias

[Top](#)

A service alias allows a service to appear in more than one folder in the Services and Policies list. The service alias is a linked copy of the original service.

Create

Creates a new service alias

Request

```
POST 1.0/serviceAliases
```

Body

The service alias to create

Element

l7:ServiceAlias

Content-Type

application/xml

Response

Body

A reference to the newly created service alias

List

Returns a list of service aliases. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/serviceAliases?service.id=26df9b0abc4dd6780fd9da5929cde13e
```

Returns service aliases for service with ID "26df9b0abc4dd6780fd9da5929cde13e".

If a parameter is not a valid search value a bad request error will be returned.

Request

```
GET 1.0/serviceAliases
```

Query Parameters

Param	Type	Default	Since	Description
sort	id service.id			Key to sort the list by

	folder.id			
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
service.id	string			Service id filter
folder.id	string			Folder id filter
securityZone.id	string			Security zone ID filter

Response

Body

A list of service aliases. If the list is empty then no service aliases were found.

Element	l7:List
----------------	---------

Delete

Deletes an existing service alias.

Request

DELETE 1.0/serviceAliases/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the service alias to delete.

Response

No Response Body

Get

Returns a service alias with the given ID.

Request

GET 1.0/serviceAliases/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the service alias to return

Response

Body

The service alias.

Element	l7:Item
----------------	---------

Update Or Create

Creates or Updates an existing service alias. If a service alias with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/serviceAliases/{id}

Path Parameters

Param	Type	Description
id	string	ID of the service alias to create or update

Body

Service alias to create or update

Element	l7:ServiceAlias
Content-Type	application/xml

Response

Body

A reference to the newly created or updated service alias.

Template

Returns a template, which is an example service alias that can be used as a reference for what service alias objects should look like.

Request

GET 1.0/serviceAliases/template

Response

Body

The template service alias.

Element	l7:Item
----------------	---------

Site Minder Configuration

Create

Creates a new SiteMinder configuration

Request

POST 1.0/siteMinderConfigurations

Body

The SiteMinder configuration to create

Element	I7:SiteMinderConfiguration
Content-Type	application/xml

Response

Body

A reference to the newly created SiteMinder configuration

List

Returns a list of SiteMinder configurations. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

Examples:

```
/restman/1.0/siteMinderConfigurations?name=MySiteMinderConfiguration
```

Returns SiteMinder configuration with name "MySiteMinderConfiguration".

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/siteMinderConfigurations

Query Parameters

Param	Type	Default	Since	Description
sort	id name			Key to sort the list by
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
name	string			Name filter
enabled	boolean			Enabled filter

securityZone.id	string			Security zone ID filter
-----------------	--------	--	--	-------------------------

Response

Body

A list of SiteMinder configurations. If the list is empty then no SiteMinder configurations were found.

Element	I7:List
----------------	---------

Delete

Deletes an existing SiteMinder configuration.

Request

DELETE 1.0/siteMinderConfigurations/{id}
--

Path Parameters

Param	Type	Description
id	string	The ID of the SiteMinder configuration to delete.

Response

No Response Body

Get

Returns a SiteMinder configuration with the given ID.

Request

GET 1.0/siteMinderConfigurations/{id}

Path Parameters

Param	Type	Description
id	string	The ID of the SiteMinder configuration to return

Response

Body

The SiteMinder configuration

Element	I7:Item
----------------	---------

Update

Creates or Updates an existing SiteMinder configuration. If a SiteMinder configuration with the given ID does not exist one will be created, otherwise the existing one will be updated.

Request

PUT 1.0/siteMinderConfigurations/{id}

Path Parameters

Param	Type	Description
id	string	ID of the SiteMinder configuration to create or update

Body

SiteMinder configuration to create or update

Element	I7:SiteMinderConfiguration
Content-Type	application/xml

Response

Body

A reference to the newly created or updated SiteMinder configuration.

Template

Returns a template, which is an example SiteMinder configuration that can be used as a reference for what SiteMinder configuration objects should look like.

Request

GET 1.0/siteMinderConfigurations/template

Response

Body

The template SiteMinder configuration.

Element	I7:Item
----------------	---------

User

[Top](#)

A user represents a user identity in an identity provider. When no identity provider is specified in the url then the internal identity provider is assumed. Users can only be created and updated in the internal identity provider.

Create User

Creates a new user. New users can only be created on the internal identity provider.

Request

POST 1.0/users

Body

The user to create.

Element	I7:User
Content-Type	application/xml

Response

Body

A reference to the newly created user

List Users

Returns a list of users. Can optionally sort the resulting list in ascending or descending order. Other params given will be used as search values.

If a parameter is not a valid search value a bad request error will be returned.

Request

GET 1.0/users

Query Parameters

Param	Type	Default	Since	Description
sort	id login			Key to sort the list by.
order	asc desc			Sort order for the list; 'asc'=ascending, 'desc'=descending; defaults to ascending if not specified
login	string			Login filter

Response

Body

A list of groups. If the list is empty then no groups were found.

Element	I7:List
----------------	---------

Delete User

Deletes an existing user

Request

```
DELETE 1.0/users/{userID}
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user to delete.

Response

No Response Body

Get User

Returns a user with the given ID.

Request

```
GET 1.0/users/{userID}
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user to return

Response

Body

The user.

Element	l7:Item
----------------	---------

Update User

Updates an existing user

Request

```
PUT 1.0/users/{userID}
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user to update

Body

The updated user

Element	I7:User
Content-Type	application/xml

Response

Body

A reference to the newly updated user.

Delete User Certificate

Removes the certificate from the user

Request

DELETE 1.0/users/{userID}/certificate

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Response

No Response Body

Get User Certificate

Gets the user's certificate

Request

GET 1.0/users/{userID}/certificate

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Response

Body

The certificate

Element	I7:Item
----------------	---------

Set User Certificate

Set this user's certificate

Request

```
PUT 1.0/users/{userID}/certificate
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Body

The certificate data

Element	l7:CertificateData
Content-Type	application/xml

Response

Body

The certificate set on the user

Element	l7:Item
----------------	---------

Change User Password

Change this user's password

Request

```
PUT 1.0/users/{userID}/password
```

Path Parameters

Param	Type	Description
userID	string	The ID of the user

Query Parameters

Param	Type	Default	Since	Description
format	string	plain		The format of the password. "plain" or "sha512crypt"

Body

The new password

Element	xsd:string
----------------	------------

Content-Type	application/xml
---------------------	-----------------

Response

Body

The user that the password was changed for.

Element	l7:Item
----------------	---------

Template User

Returns a template, which is an example user that can be used as a reference for what user objects should look like.

Request

GET 1.0/users/template

Response

Body

The template user.

Element	l7:Item
----------------	---------

