

REST API upgrade



Version warning

In Syncope 1.1 there are two REST interfaces:

1. a new one (referred as **new** in the following) exposed under `/cxf`
2. the one available in 1.0 (referred as **old** in the following) is still present but will be removed from releases $\geq 1.2.0$. It is exposed by default in 1.1 under `/rest`

In Syncope ≥ 1.2 there is a single REST interface, available under `/rest`

Starting with Syncope 2.0 the [Reference Guide](#) is available.

This page shall give you an overview of old and new REST API.

- [Configuration Service](#)
- [Connector Service](#)
- [Entitlement Service](#)
- [Logger Service](#)
- [Notification Service](#)
- [Policy Service](#)
- [Report Service](#)
- [Resource Service](#)
- [Role Service](#)
- [Schema Service](#)
- [Task Service](#)
- [User Service](#)
- [UserWorkflow Service](#)
- [UserRequest Service](#)
- [Workflow Service](#)

Main focus on redesign REST interface was to apply [RESTful Best Practices](#)

- use HTTP operations instead of URL encoded operation names
- GET does not modify any object (read-only safety operation)
- PUT and DELETE are idempotent operations
 - PUT and DELETE do not return (old) object any longer, except for Roles and Users, since they contain propagation status information
- CREATE operations return location URL in header for newly created object
 - In addition to that unique identifier of resource is also send in response header (`org.apache.syncope.resource.id`)

While redesigning interface, we also made changes to [exception handling](#).

Configuration Service

Old URL	New URL	Comment
POST /configuration/create	POST /configurations	Creates a new Configuration.
GET /configuration/read/{key}	GET /configurations/{key}	Returns configuration element with matching key.
GET /configuration/list	GET /configurations	Returns a list of all configuration elements.
POST /configuration/update	PUT /configurations/{key}	Overwrites configuration element with matching key.
GET /configuration/delete/{key}	DELETE /configurations/{key}	Deletes configuration with matching key.
Old URL	New URL	Comment
GET /configuration/validators	GET /configurations/validators	Returns a list of known validators.
GET /configuration/mailTemplates	GET /configurations/mailTemplates	Returns a list of known mail-templates.
GET /configuration/dbexport	GET /configurations/stream	Returns configuration as an downloadable content.xml database export file.

Connector Service

Old URL	New URL	Comment
POST /connector/create	POST /connectors	Creates a new connector instance.

GET /connector/read/{connectorId}	GET /connectors/{connectorId}	Returns connector with matching id.
GET /connector/list?lang={lang}	GET /connectors?lang={lang}	Returns a list of all connectors. Default language is English.
POST /connector/update	PUT /connectors/{connectorId}	Overwrites connector with matching key.
GET /connector/delete/{connectorId}	DELETE /connectors/{connectorId}	Deletes connector with matching id.
Old URL	New URL	Comment
GET /connector/bundle/list?lang={lang}	GET /connectors/bundles?lang={lang}	Returns known bundles. Default language is English.
POST /connector/schema/list?showall={showall}	POST /connectors/{connectorId}/schemas?showAll={showall}	Returns schema names for connector. Default is showAll=false.
GET /connector/{connectorId}/configurationProperty/list	GET /connectors/{connectorId}/configuration	Returns configuration for selected connector.
POST /connector/check	POST /connectors/check	Checks if a connection can be established.
GET /connector/{resourceName}/readByResource	GET /connectors;resourceName={connectorId}	Returns connector for resourceName.
POST /connector/reload	POST /connectors/reload	Reload all connector bundles and instances.

Entitlement Service

Old URL	New URL	Comment
GET /auth/allentitlements	GET /entitlements	Returns a list of all known entitlements.
GET /auth/entitlements	GET /entitlements/own	Returns a list of entitlements assigned to the authenticated user.

Logger Service

Old URL	New URL	Comment
POST /logger/log/{name}/{level}	PUT /logger/normal/{name}	Creates or updates a logger with given name and sets logging level. Level will be now be part of payload instead of URL parameter.
GET /logger/log/list	GET /logger/normal	Returns a list of all normal logger.
GET /logger/delete/{name}	DELETE /logger/normal/{name}	Deletes normal logger with matching name.
Old URL	New URL	Comment
PUT /logger/audit/enable	PUT /logger/audit/{name}	Creates or updates audit logger with matching name.
GET /logger/audit/list	GET /logger/audit	Returns a list of all audit logger.
PUT /logger/audit/disable	DELETE /logger/audit/{name}	Deletes audit logger with matching name.

Notification Service

Old URL	New URL	Comment
POST /notification/create	POST /notifications	Creates a new notification.
GET /notification/read/{notificationId}	GET /notifications/{notificationId}	Returns a notification with matching id.
GET /notification/list	GET /notifications	Returns a list of all notifications.
POST /notification/update	PUT /notifications/{notificationId}	Updates notification with matching id.
GET /notification/delete/{notificationId}	DELETE /notifications/{notificationId}	Deletes notification with matching id.

Policy Service

Old URL	New URL	Comment
POST /policy/account/create	POST /policies/account	Creates a new account policy.
GET /policy/read/{id}	GET /policies/account/{id}	Returns account policy with matching id.
GET /policy/account/global/read	GET /policies/account/0	Returns global account policy.
GET /policy/account/list	GET /policies/account	Returns a list of account policies.
POST /policy/account/update	PUT /policies/account/{id}	Updates account policy with matching id.
GET /policy/delete/{id}	DELETE /policies/account/{id}	Deletes account policy with matching id.
Old URL	New URL	Comment
POST /policy/sync/create	POST /policies/sync	Creates a new sync policy.
GET /policy/read/{id}	GET /policies/sync/{id}	Returns sync policy with matching id.
GET /policy/sync/global/read	GET /policies/sync/0	Returns global sync policy.
GET /policy/sync/list	GET /policies/sync	Returns a list of sync policies.
POST /policy/sync/update	PUT /policies/sync/{id}	Updates sync policy with matching id.
GET /policy/delete/{id}	DELETE /policies/sync/{id}	Deletes sync policy with matching id.
Old URL	New URL	Comment
POST /policy/password/create	POST /policies/password	Creates a new password policy.
GET /policy/read/{id}	GET /policies/password/{id}	Returns password policy with matching id.
GET /policy/password/global/read	GET /policies/password/0	Returns global password policy.
GET /policy/password/list	GET /policies/password	Returns a list of password policies.
POST /policy/password/update	PUT /policies/password/{id}	Updates password policy with matching id.
GET /policy/delete/{id}	DELETE /policies/password/{id}	Deletes password policy with matching id.

Report Service

Old URL	New URL	Comment
POST /report/create	POST /reports	Creates a new report.
GET /report/read/{reportId}	GET /reports/{reportId}	Returns report with matching reportId.
GET /report/list	GET /reports	Returns a list of all reports.
GET /report/list/{page}/{size}	GET /reports?page={page}&size={size}	Returns a list of reports according to pagination.
POST /report/count	POST /reports/count	Returns number of existing reports.
POST /report/update	PUT /reports/{reportId}	Updates report with matching reportId.
GET /report/delete/{reportId}	DELETE /reports/{reportId}	Deletes report with matching id.
Old URL	New URL	Comment
POST /report/execute/{reportId}	POST /reports/{reportId}/execute	Executes report and returns execution result.
GET /report/execution/read/{executionId}	GET /reports/executions/{executionId}	Returns execution report.
GET /report/execution/export/{executionId}?fmt={format}	GET /reports/executions/{executionId}/dbDump?format={format}	Returns execution report as an downloadable file. Format is optional.
GET /report/execution/delete/{executionId}	DELETE /reports/executions/{executionId}	Deletes execution report with matching id.
Old URL	New URL	Comment

GET /report/reportletConfClasses	GET /reports/reportletConfClasses	Returns a list of all reportletConfClasses.
----------------------------------	-----------------------------------	---

Resource Service

Old URL	New URL	Comment
POST /resource/create	POST /resources	Creates a new resource.
GET /resource/read/{resourceName}	GET /resources/{resourceName}	Returns a resource with specified name.
GET /resource/list	GET /resources	Returns a list of all resources.
GET /resource/list?connInstanceId={connInstanceId}	GET /resources;connectorId={connectorId}	Returns a list of all resources used by matching connector.
POST /resource/update	PUT /resources/{resourceName}	Updates resource with matching resourceName.
GET /resource/delete/{resourceName}	DELETE /resources/{resourceName}	Deletes resource with specified resourceName.
Old URL	New URL	Comment
GET /resource/{resourceName}/read/{type}/{id}	GET /resources/{resourceName}/{type}/{id}	Returns connector object from the external resource, for the given type and id.
POST /resource/check	POST /resources/check	Verifies connection using resource connector parameters.
GET /resource/propagationActionsClasses	GET /resources/propagationActionsClasses	Returns a list of classes with name PROPAGATION_ACTIONS.

Role Service

Old URL	New URL	Comment
POST /role/create	POST /roles/	Creates a new role. Returns URL to new role definition
GET /role/read/{roleId}	GET /roles/{roleId}	Returns a single role definition matching the provided roleId
GET /role/selfRead/{roleId}	GET /roles/{roleId}	Authorization mechanisms will be independent of URL
GET /role/list	GET /roles	Returns a list of all known roles
GET /role/parent/{roleId}	GET /roles/{roleId}/parent	Returns a single parent role definition (if available)
GET /role/children/{roleId}	GET /roles/{roleId}/children	Returns a list of children role definitions
POST /role/update	POST /roles/{roleId}	Updates role. (POST is used here instead of PUT, because RoleMod only contains changes for a role and not a complete representation thereof.)
GET /role/delete/{roleId}	DELETE /roles/{roleId}	Deletes role.

Schema Service

Info: {kind} can be USER, ROLE or MEMBERSHIP.

Old URL	New URL	Comment
POST /schema/{kind}/create	POST /schemas/{kind}/NORMAL	Creates a new "normal" schema for matching kind.
GET /schema/{kind}/read/{name}	GET /schemas/{kind}/NORMAL/{name}	Returns matching schema.
GET /schema/{kind}/list	GET /schemas/{kind}/NORMAL	Returns a list of all "normal" schema definitions of matching kind.
POST /schema/{kind}/update	PUT /schemas/{kind}/NORMAL/{name}	Updates matching schema.
GET /schema/{kind}/delete/{name}	DELETE /schemas/{kind}/NORMAL/{name}	Deletes matching schema.
Old URL	New URL	Comment
POST /virtualSchema/{kind}/create	POST /schemas/{kind}/VIRTUAL	Creates a new "virtual" schema for matching kind.

GET /virtualSchema/{kind}/read/{name}	GET /schemas/{kind}/VIRTUAL/{name}	Returns matching schema.
GET /virtualSchema/{kind}/list	GET /schemas/{kind}/VIRTUAL	Returns a list of all "virtual" schema definitions of matching kind.
POST /virtualSchema/{kind}/update	PUT /schemas/{kind}/VIRTUAL/{name}	Updates matching schema.
GET /virtualSchema/{kind}/delete/{name}	DELETE /schemas/{kind}/VIRTUAL/{name}	Deletes matching schema.
Old URL	New URL	Comment
POST /derivedSchema/{kind}/create	POST /schemas/{kind}/DERIVED	Creates a new "derived" schema for matching kind.
GET /derivedSchema/{kind}/read/{name}	GET /schemas/{kind}/DERIVED/{name}	Returns matching schema.
GET /derivedSchema/{kind}/list	GET /schemas/{kind}/DERIVED	Returns a list of all "derived" schema definitions of matching kind.
POST /derivedSchema/{kind}/update	PUT /schemas/{kind}/DERIVED/{name}	Updates matching schema.
GET /derivedSchema/{kind}/delete/{name}	DELETE /schemas/{kind}/DERIVED/{name}	Deletes matching schema.

Task Service

Old URL	New URL	Comment
POST /task/create/sync	POST /tasks	Creates a new sync task.
GET /task/read/{taskId}	GET /tasks/sync/{taskId}	Returns sync task with matching id.
GET /task/sync/list	GET /tasks/sync/list	Returns a list of sync tasks.
GET /task/sync/list/{page}/{size}	GET /tasks/sync?page={page}&size={size}	Returns a list of paginated sync tasks.
GET /task/sync/count	GET /tasks/sync/count	Returns number of sync tasks.
POST /task/update/sync	PUT /tasks/{taskId}	Updates sync task with matching id.
Old URL	New URL	Comment
POST /task/create/sched	POST /tasks	Creates a new sched task.
GET /task/read/{taskId}	GET /tasks/sched/{taskId}	Returns sched task with matching id.
GET /task/sched/list	GET /tasks/sched/list	Returns a list of sched tasks.
GET /task/sched/list/{page}/{size}	GET /tasks/sched?page={page}&size={size}	Returns a list of paginated sched tasks.
GET /task/sched/count	GET /tasks/sched/count	Returns number of sched tasks.
POST /task/update/sched	PUT /tasks/{taskId}	Updates sched task with matching id.
Old URL	New URL	Comment
GET /task/read/{taskId}	GET /tasks/propagation/{taskId}	Returns propagation task with matching id.
GET /task/propagation/list	GET /tasks/propagation/list	Returns a list of propagation tasks.
GET /task/propagation/list/{page}/{size}	GET /tasks/propagation?page={page}&size={size}	Returns a list of paginated propagation tasks.
GET /task/propagation/count	GET /tasks/propagation/count	Returns number of propagation tasks.
GET /task/propagation/execution/list	GET /tasks/propagation/executions	Returns list of propagation task executions.
Old URL	New URL	Comment
GET /task/read/{taskId}	GET /tasks/notification/{taskId}	Returns notification task with matching id.
GET /task/notification/list	GET /tasks/notification/list	Returns a list of notification tasks.
GET /task/notification/list/{page}/{size}	GET /tasks/notification?page={page}&size={size}	Returns a list of paginated notification tasks.

GET /task/notification/count	GET /tasks/notification/count	Returns number of notification tasks.
Old URL	New URL	Comment
GET /task/delete/{taskId}	DELETE /tasks/{taskId}	Deletes task with matching id.
GET /task/execution/read/{executionId}	GET /tasks/executions/{executionId}	Returns execution of task with matching id.
POST /task/execute/{taskId}?dryRun={dryRun}	POST /tasks/{taskId}/execute?dryRun={dryRun}	Executes task with matching id.
GET /task/execution/delete/{executionId}	DELETE /tasks/executions/{executionId}	Deletes execution of task with matching id.
GET /task/jobClasses	GET /tasks/jobClasses	Returns a list of available jobClasses.
GET /task/syncActionsClasses	GET /tasks/syncActionsClasses	Returns a list of available syncActionsClasses.
GET /task/execution/report/{executionId} ?executionStatus={executionStatus}&message={message}	POST /tasks/executions/{executionId}/report	Creates a new report for matching execution id.

User Service



Version warning

User Service Interface in Version 1.1.0 is still not applying some major RESTful best practices. Therefore refactoring for release $\geq 1.2.0$ is most likely.

CRUD operations:

Old URL	New URL	Comment
POST /user/create	POST /users	Creates a new user.
GET /user/list	GET /users	Returns a list of all known users
GET /user/list/{page}/{size}	GET /users?page={page}&size={size}	Returns a list of known users matching page/size conditions
GET /user/read/{userId}	GET /users/{userId}	Returns a single user matching the provided userId
GET /user/read/self	GET /users/self	Returns the authenticated user
GET /user/readByUsername/{username}	GET /users?username={username}	Returns a single user matching the provided username
POST /user/search	POST /users/search	Returns a list of user matching the provided search conditions.
POST /user/search/{page}/{size}	POST /users/search?page={page}&size={size}	Returns a list of user matching the provided page/size and search conditions.
GET /user/count	GET /users/count	Returns a number of existing users.
POST /user/search/count	POST /users/search/count	Returns a number of user matching the provided search conditions.
GET /user/verifyPassword/{username}?password={password}	<<removed>>	Use Basic-Authentication and GET /users/self to verify authentication and retrieve user object including assigned roles.
POST /user/update	POST /users/{userId}	Updates user. (POST is used here instead of PUT, because UserMod only contains changes for a user and not a complete representation thereof.)
GET /user/delete/{userId}	DELETE /users/{userId}	Deletes user.
GET /deleteByUsername/{username}	<<removed>>	Please use 'GET /user?username={username}' to discover userId and then use 'DELETE /user/{userId}' to delete user.

Account status:

Old URL	New URL	Comment
GET /user/activate/{userId}	POST /users/{userId}/status/activate	Activates matching user account.
GET /user/activateByUsername/{username}	POST /user/activateByUsername/{username}	Activates matching user account.

GET /user/reactivate/{userId}	POST /users/{userId}/status/reactivate	Reactivates new user account.
GET /user/reactivateByUsername/{username}	POST /user/reactivateByUsername/{username}	Reactivates new user account.
GET /user/suspend/{userId}	POST /users/{userId}/status/suspend	Suspends user account.
GET /user/suspendByUsername/{username}	POST /user/suspendByUsername/{username}	Suspends user account.

UserWorkflow Service

Old URL	New URL	Comment
GET /user/workflow/form/list	GET /userworkflow/forms	Returns a list of user workflow forms.
POST /user/workflow/form/submit	POST /userworkflow/forms	Submits a form to a user workflow.
GET /user/workflow/form/{userId}	GET /userworkflow/forms/{userId}	Returns a (single) workflow form assigned to user.
POST /user/execute/workflow/{taskId}	POST /userworkflow/tasks/{taskId}/execute	Executes workflow task on user.
GET /workflow/form/claim/{taskId}	POST /userworkflow/tasks/{taskId}/claim	Claims workflow task for authenticated (session) user.

UserRequest Service

Old URL	New URL	Comment
POST /user/request/create	POST /requests/user	POST Content contains user and create status.
POST /user/request/update	POST /requests/user	POST Content contains userMod and update status.
GET /user/request/delete/{userId}	POST /requests/user	POST Content contains user id and delete status.
GET /user/request/list	GET /requests/user	Returns a list of all user requests.
GET /user/request/read/{requestId}	GET /requests/user/{requestId}	Returns a user request with matching id.
GET /user/request/deleteRequest/{requestId}	DELETE /requests/user/{requestId}	Deletes request with matching id.
Old URL	New URL	Comment
GET /user/request/create/allowed	GET /requests/user/create/allowed	Returns true/false whether user is allowed to create a new user.
GET /user/request/create/allowed	OPTIONS	Returns HTTP header "Syncope-Create-Allowed" indicating if user create is allowed or not.

Workflow Service

Old URL	New URL	Comment
PUT /workflow/definition/user	PUT /workflows/user	Creates or updates workflow definition for users.
GET /workflow/definition/user	GET /workflows/user	Returns a workflow definition for users.
GET /workflow/tasks/user	GET /workflows/user/tasks	Returns workflow tasks for users.
Old URL	New URL	Comment
PUT /workflow/definition/role	PUT /workflows/role	Creates or updates workflow definition for roles.
GET /workflow/definition/role	GET /workflows/role	Returns a workflow definition for roles.
GET /workflow/tasks/role	GET /workflows/role/tasks	Returns workflow tasks for roles.