

Security and OAuth 2.0

Protect the inbound API, restrict the objects each SAP identity may use, and separate operational permissions from execution rights.

Apply each security layer

- Use HTTPS with a valid certificate for every API request. Basic credentials and bearer tokens are secrets.
- Use a separate named SAP integration identity for each integration and verify the actual identity in the backend.
- Maintain narrow `/CGR/API_AUTH` grants in the application client. An exact user and a matching object grant are required.
- Assign the SAP roles required by the selected interface and business operations. Connector configuration is not a substitute for those roles.
- Restrict monitor payload access, result/rule administration, rerun execution, cleanup and background-job administration separately.

Authentication by interface

Interface	Supported authentication	Operational note
SICF	HTTP Basic only, over HTTPS	Send the SAP username and password; confirm the intended application client.
Gateway OData	Any OAuth 2.0 flow supported by the SAP system	Choose the flow available in your installed SAP release and configure its client, identity and service scopes.

Client Credentials is one OAuth example, not a requirement imposed by the Connector. Do not assume every SAP release offers the same grants or administrative screens. Basis must use the flow and role setup supported by the target installation.

Configure an OAuth client for OData

1. Register and activate `/CGR/CONN_SRV` in `/IWFND/MAINT_SERVICE` for the intended system alias and enable its OAuth protection.
2. Use the SAP OAuth client configuration appropriate to your release, such as `SOAUTH2 / oauth2_config`. Register the client and chosen supported grant; bind the appropriate SAP integration identity.
3. Assign the generated service scope. Where `S_SCOPE` is used, maintain `OA2_CLIENT` and `OA2_SCOPE` in the SAP role and assign it to the appropriate identity.
4. Add the exact integration user and permitted objects to `/CGR/API_AUTH` in the application client. Assign backend roles for the actual requested operations.
5. Obtain a token with the selected flow, then verify one allowed request and one denied request using fresh authenticated sessions.

Client Credentials example

```
POST /sap/bc/sec/oauth2/token HTTP/1.1
Host: sap.example.com
Authorization: Basic <base64(client_id:client_secret)>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=<url-encoded-service-scope>
```

Use the actual generated scope and configured token endpoint. Send the returned token with API requests as Authorization: Bearer <access_token>. Token lifetimes and renewal follow SAP configuration; do not embed credentials or tokens in source code or URL query values.

Object grants and metadata

An exact user is required in /CGR/API_AUTH. Object Name can be exact or a complete *. API BO can be * only with Object Name *. Partial patterns such as BAPI_* and wildcard users are not supported. A broad existing grant still applies after a narrow row is removed; there are no deny overrides.

Metadata requests are read-only but still check license, object grants and the active backend definition. Restrict them as you restrict execution. Do not expose internal object inventories based solely on a valid HTTP login.

Network and identity behavior

CGR Connector initiates no outbound HTTP or HTTPS calls. There are no HTTP/HTTPS license checks, update checks or telemetry. The SICF option can execute through an approved backend RFC route. Gateway OData uses the backend already selected by Gateway and requires no additional Connector RFC routing.

The current SICF execution policy rejects a destination that maps the request to a different technical user. The monitor records observed execution identity. Caller IP is the observed connection peer; behind a proxy it can be the proxy address. Forwarding headers are diagnostic context, not proof of end-user identity.

Optional email uses native SAPconnect submission. Basis configures its SMTP route. Notification senders and recipients are explicit configuration; email is disabled after basic installation until capture, preparation and sending are deliberately enabled.

Operational permissions

Action	Permission boundary
Monitor display	Authorization object /CGR/MONI, activity 03
Manual FUNCTION rerun	Current operator display plus activity 16; source evidence, target and license/grants are rechecked
Correct draft input	Activity 02 in addition to the rerun permissions
Versioned cleanup	Display plus administer activity 70 and the tool-specific Execute controls
Result/rule administration	Dedicated S_TABU_NAM authority for the relevant configuration table and expected-revision publication

A historical user name in a log is not permission to execute as that user. Manual reruns execute as the current operator in the same backend/client. Original inputs and results remain immutable; corrected input

is recorded on a new attempt.

Handle retained data deliberately

Request URLs, payloads, response data and captured email content may contain personal or sensitive business data. Agree retention and authorized access before production use. Payload expiry can precede header expiry; notification retention is independent of log retention. Avoid secrets in business payloads and query strings, and send support only the minimal redacted evidence needed.

CGR SOFTWARE

cgrsoft.com · sales@cgrsoft.com

Use these release 2.0 instructions with the supplied release delivery notes. Release 1.3 has a separate documentation and PDF set.