



Security And OAuth 2.0

Authentication establishes who is calling. CGR whitelisting and SAP authorization determine what they may reach. Apply all three layers.



Authentication establishes caller identity; CGR whitelisting and SAP authorization determine access permissions. Apply all three security layers for comprehensive protection.

-  **Transport security.** Activate SSL and use HTTPS exclusively.
-  **Identity isolation.** Deploy a separate SAP identity for each external integration, rather than sharing one.
-  **CGR authorization.** Configure the permitted objects in `/CGR/API_AUTH`.
-  **Backend authorization.** Preserve the standard SAP checks for tables, RFC functions and OAuth scopes.
-  **Auditability.** Use `/CGR/MONI` and `/CGR/API_LOG` for request and response tracing.



BASELINE RECOMMENDATIONS

Implement HTTPS, use dedicated technical identities, apply the narrowest possible CGR object whitelists, and maintain narrowly scoped SAP roles. Store credentials securely outside source code, and rotate them to your own policy.

No outbound HTTP/HTTPS calls

CGR SAP Connector does not initiate outbound HTTP or HTTPS calls. This includes license checks, update checks and telemetry.

Backend RFC connections remain supported for the SICF option. The OData option does not require a backend RFC connection.



The SICF option supports HTTP Basic authentication only. Send an SAP username and password with each request over HTTPS.

- Deploy restricted technical users rather than personal dialog accounts.
- Restrict object access strictly in `/CGR/API_AUTH`.
- Assign only the necessary SAP table, report and RFC authorizations.
- Mandate HTTPS, to prevent credential exposure over unencrypted channels.



AUTHENTICATION BY CONNECTOR OPTION

SICF supports HTTP Basic only. OData supports any OAuth 2.0 flow supported by your SAP system; Client Credentials is one example.



The OData option supports any OAuth 2.0 flow supported by your SAP system. The steps below show Client Credentials as one example.

- 1** Create a System-type user in `SU01` (for example `0AUTH_CLIENT_01`), configure a secure secret, keep it active, and add it to `/CGR/API_AUTH`.
- 2** Enable OAuth on the OData service via `/IWFND/MAINT_SERVICE`, and confirm the OAuth indicator appears.
- 3** Register the client at the OAuth configuration application, reachable via `S0AUTH2`. Ensure the client ID matches the system user.
- 4** Add the service-generated scope and select the **Client Credentials** grant type.

```
https://<server>:<port>/sap/bc/webdynpro/sap/oauth2_config?sap-client=<client>&sap-language=EN#
```

Prerequisites for this example

A functional SAP NetWeaver Gateway, HTTPS/SSL configured in STRUST, and administrative access to Gateway security transactions.



- 1 Open `PFCG` and create a role such as `Z_OAUTH_ODATA_ACCESS`, or update an existing one.
- 2 Manually add authorization object `S_SCOPE`.
- 3 Set `OA2_CLIENT` to the registered client ID (for example `OAUTH_CLIENT_01`).
- 4 Set `OA2_SCOPE` to the generated service scope (for example `/CGR/CONN_SRV_0001`).
- 5 Generate the profile and assign the role to the identity used for OAuth-protected calls.



A VALID BEARER TOKEN IS NOT ENOUGH

The SAP identity still requires both its `/CGR/API_AUTH` configuration and the standard backend authorizations for the objects requested. Authentication and authorization remain separate questions.

TRANSACTIONS USED

Transaction	Purpose
<code>SU01</code>	Maintain the OAuth technical user
<code>/IWFND/MAINT_SERVICE</code>	Register the OData service; enable the OAuth scope
<code>SOAUTH2</code> / <code>oauth2_config</code>	Register clients; select grant types; assign scopes
<code>PFCG</code>	Create the role; maintain the <code>S_SCOPE</code> authorization object
<code>/CGR/API_AUTH</code>	Whitelist the accessible SAP tables, functions and reports

CGR SOFTWARE

Send this document to your security and Basis teams.

cgrsoft.com

sales@cgrsoft.com