Berechtigungsverwaltung

Die COBI.wms Management-Datenbank kann in **On-Premises- oder Private-Cloud-Umgebungen** installiert werden, in denen Sie direkten Zugriff auf den Datenbankserver haben und dort eigene Datenbanken erstellen können. Die Management-Datenbank ermöglicht es Ihnen, zentral Verbindungen zu SAP-Business-One-Datenbanken, COBI.wms-Benutzern und Geräten sowie Modulberechtigungen zu definieren und zu verwalten.

Erstellen der Datenbank

Wenn Sie MS SQL Server verwenden, führen Sie den Inhalt der folgenden Datei im SQL Server Management Studio aus:

cobiwms-mssql.sql

Wenn Sie SAP HANA verwenden, führen Sie stattdessen Folgendes im **HANA Studio** aus:

cobiwms-hana.sql

Firmenverbindungen

Sie müssen SAP-Business-One-Datenbankverbindungen definieren, indem Sie Datensätze in die Tabelle **companies** einfügen.

In der Regel müssen nur die folgenden Spalten ausgefüllt werden:

Spalte	Typ / Gültige Werte	Beschreibung
CompanyID	Text	Eindeutige Kennung für diese Verbindung
SQLDB	Text	Name der SAP-Business-One-Datenbank
APIType	SL oder IF	SL für Service Layer; IF für Integration Framework
APIURL	Text	Service-Layer-URL oder Integration-Framework-Trigger-URL
APIID	Text	Für SL: identisch mit SQLDB; für IF: Firmen-ID im IF-SLD
APIUsername	Text oder NULL	Für SL: SAP-Business-One-Benutzername; für IF: NULL
APIPassword	Text oder NULL	Für SL: SAP-Business-One-Passwort; für IF: NULL

Tipp: Sie können die Zeichenfolge **{host}** als Teil des APIURL-Wertes verwenden, damit die App denselben Hostnamen (oder dieselbe IP-Adresse) wie die Management-Datenbank verwendet.

Tipp: *(nur für Service Layer)* Wenn Sie möchten, dass Lagerbenutzer sich mit individuellen Logins anmelden, können Sie die Felder APIUsername und APIPassword leer lassen und stattdessen die Felder APIUser und APIPass in der Tabelle **Users** ausfüllen, wie im Abschnitt Separater Login pro Benutzer beschrieben.

Hinweis: Wenn der ApiType auf SL steht, kann die Spalte SQLDB leer bleiben. Dadurch verwendet die App eine reine Service-Layer-Verbindung (ähnlich einer Cloud-Umgebung). Dies wird jedoch **nicht empfohlen**, da eine direkte Datenbankverbindung deutlich mehr Leistung und Stabilität bietet. Die

Spalte SQLDB sollte daher bei **allen On-Premises-Installationen** ausgefüllt werden.

Weitere Informationen: Architecture Overview

Beispiel für das Hinzufügen einer **Produktiv-** und einer **Testverbindung** in einer On-Premises-Umgebung mit unverschlüsselter Service-Layer-Kommunikation (kein SSL-Zertifikat erforderlich):

```
INSERT INTO companies (companyId, sqlDb, apiType, apiUrl, apiId,
apiUsername, apiPassword) VALUES
('01 - PROD', 'SBO_PROD', 'SL', 'http://{host}:50001/b1s/v2', 'SBO_PROD',
'manager', 'secret');

INSERT INTO companies (companyId, sqlDb, apiType, apiUrl, apiId,
apiUsername, apiPassword) VALUES
('02 - TEST', 'SBO_TEST', 'SL', 'http://{host}:50001/b1s/v2', 'SBO_TEST',
'manager', 'secret');
```

Hinweis: Die obigen Beispiele verwenden **`http:`** statt **`https:`** sowie den Port **50001** statt **50000**. Dies bedeutet, dass die Kommunikation mit dem Service Layer **unverschlüsselt** erfolgt und die App den **Load Balancer überspringt**, um direkt **Node 1** des Service Layers anzusprechen.

Wenn Sie **verschlüsselte Kommunikation** mit dem Service Layer wünschen oder **Performanceprobleme** durch das Umgehen des Load Balancers auftreten, müssen Sie sicherstellen, dass ein **gültiges SSL-Zertifikat** installiert ist und anschließend **„http“ in „https“** sowie **Port 50001 in 50000** ändern.

Optionale Spalten

Die folgenden Spalten der Tabelle **companies** sollten in der Regel leer bleiben (`NULL` oder leerer Text):

Spalte	Typ / Gültige Werte	Beschreibung
DBType	MSSQL oder HANA	MSSQL für SQL Server, HANA für SAP HANA
SQLHost	Text	Hostname oder IP-Adresse des Datenbankservers
SQLPort	Text	Portnummer des Datenbankservers
SQLUser	Text	Datenbank-Benutzername (z. B. 'sa' oder 'SYSTEM')
SQLPass	Text	Datenbank-Passwort
SQLDomain	Text	Domäne für „Trusted Connection“ im SQL Server
HANAProxyHost	Text	Hostname oder IP-Adresse des HANA Proxy
HANAProxyPort	Text	Portnummer des HANA Proxy
Profile	Text	Aktiviert ein kundenspezifisches Profil
PrintService	Text	Adresse des COBI.wms Print Service

Die Spalten `DBType`, `SQLHost`, `SQLUser` und `SQLPass` müssen nur ausgefüllt werden, wenn sich die SAP-Business-One-Datenbank **auf einem anderen Server** befindet als die Management-Datenbank. Die Spalte `SQLPort` wird nur benötigt, wenn der Server nicht den Standardport verwendet (1433 für MS SQL Server, 30015 für SAP HANA).

`HANAProxyHost` muss nur gefüllt werden, wenn der Proxy nicht auf demselben Server wie die HANA-Datenbank läuft. `HANAProxyPort` nur, wenn der Proxy einen anderen Port als den Standardwert 30075 nutzt.

Die Spalte `Profile` wird verwendet, um kundenspezifische Anpassungen zu aktivieren, und sollte leer bleiben, sofern nichts anderes angegeben wurde.

Die Spalte `PrintService` kann genutzt werden, um zentral die Adresse des COBI.wms Print Service zu definieren. Wenn sie hier nicht gesetzt ist, muss sie direkt auf jedem Android-Gerät in den Print Settings der App hinterlegt werden. Bei Verwendung des Standardports genügt die Angabe des Hostnamens oder der IP-Adresse. Bei einem abweichenden Port geben Sie den Wert im Format `HOST:PORT` ein.

Geräte und Benutzer

COBI.wms-Geräte

Geräte registrieren sich automatisch, sobald sie eine Verbindung zur Management-Datenbank herstellen.

Jedes Gerät erhält eine **numerische ID** beginnend bei 1. Diese ist im Login-Bildschirm der App unten rechts unterhalb des Login-Buttons sichtbar.

Ein Gerät kann über die Prozedur **`removeDevice`** entfernt werden:

```
-- MS SQL Server
EXEC removeDevice 1;

-- SAP HANA
CALL removeDevice(1);
```

COBI.wms-Benutzer

Dieser Abschnitt ist **optional** – die App kann auch **ohne definierte COBI.wms-Benutzer** in der Management-Datenbank verwendet werden.

Sie können jedoch COBI.wms-Benutzer hinzufügen, um den Zugriff auf die App genauer zu steuern und nachzuvollziehen, **welcher Benutzer welche Aktionen** ausgeführt hat. Die Felder „Username“ und „Password“ erscheinen in der App nur, wenn mindestens ein COBI.wms-Benutzer vorhanden ist.

Zum Hinzufügen eines Benutzers verwenden Sie die Prozedur **`addUser`**:

```
-- MS SQL Server
EXEC addUser 'user1', 'password', NULL, NULL, 'Full Name';

-- SAP HANA
CALL addUser('user1', 'password', NULL, NULL, 'Full Name');
```

Der Erste Parameter: Benutzer-ID (gleichzeitig der Login-Name). Kann z. B. „alice“, „bob“, „manager“ oder „produktion1“ sein.

Zweiter Parameter: Passwort. Darf nicht NULL sein, kann aber ein leerer String sein (`''`) – dann ist kein Passwort erforderlich.

Dritter und vierter Parameter: Veraltet, bitte `NULL` verwenden.

Letzter Parameter: Vollständiger Name oder Beschreibung des Benutzers, kann auch `NULL` sein.

Das Passwort eines Benutzers kann mit der Prozedur `resetPassword` zurückgesetzt werden:

```
-- MS SQL Server
EXEC resetPassword 'user1', 'new password';

-- SAP HANA
CALL resetPassword('user1', 'new password');
```

Benutzer können mit `removeUser` gelöscht werden:

```
-- MS SQL Server
EXEC removeUser 'user1';

-- SAP HANA
CALL removeUser('user1');
```

Separater Login pro Benutzer

Sie können für jeden COBI.wms-Benutzer oder jedes Gerät einen **separaten SAP-Business-One-Login** hinterlegen. Dadurch kann im SAP-Änderungsprotokoll nachvollzogen werden, **welcher Benutzer oder welches Gerät** einen Beleg gebucht oder geändert hat.

Zum Hinterlegen des Logins führen Sie folgenden SQL-Befehl in der Management-Datenbank aus:

```
UPDATE users
SET apiUser = 'sbo_username',
    apiPass = 'sbo_password'
WHERE userId = 'cobiwms_username';
```

Wenn Sie keine COBI.wms-Benutzer verwenden, können Sie denselben Ansatz für Geräte nutzen. Jedes registrierte Gerät erstellt automatisch einen speziellen COBI.wms-Benutzer namens `_deviceXXXX` (wobei XXXX die Geräte-ID ist). Sie können diesen Benutzern individuelle SAP-Business-One-Logins zuweisen:

```
UPDATE users
SET apiUser = 'wms0001',
    apiPass = 'password'
WHERE userId = '_device0001';
```

```
UPDATE users
SET apiUser = 'wms0002',
apiPass = 'password'
WHERE userId = '_device0002';
-- usw.
```

Nachdem Sie die **users**-Tabelle entsprechend aktualisiert haben, starten Sie die COBI.wms-App neu. Bei der nächsten Anmeldung wird die Änderung aktiv. Sie können eine Testbuchung durchführen und anschließend im SAP-Änderungsprotokoll prüfen, ob der korrekte Benutzer aufgeführt wird.

Warnung: Wenn Sie das SAP-Business-One-Passwort im Feld `apiPass` speichern, wird dieses **im Klartext** in der Management-Datenbank abgelegt – ebenso wie das Feld `apiPassword` in der Tabelle `companies`. Dies stellt in der Regel kein Risiko dar, solange keine unbefugten Personen Zugriff auf den Server haben. Falls dies dennoch ein Problem für Sie darstellt, siehe den folgenden Abschnitt.

Vermeidung von Klartext-Passwörtern in der Datenbank

Normalerweise muss das Passwort eines SAP-Business-One-Benutzers entweder in der Spalte **`apiPassword`** (Tabelle `companies`) oder **`apiPass`** (Tabelle `users`) angegeben werden.

(Technischer Hinweis: Eine Verschlüsselung dieser Spalten wäre sinnlos, da der Schlüssel zum Entschlüsseln in der App enthalten sein müsste und somit ausgelesen werden könnte. Eine Speicherung als Hash ist ebenfalls nicht möglich, da das Passwort im Klartext an den Service Layer übermittelt werden muss.)*

Um dies zu vermeiden, kann folgende Strategie verwendet werden:

1. Lassen Sie die Felder `apiPassword` und `apiPass` leer.
2. Erstellen Sie COBI.wms-Benutzer mit denselben Benutzernamen und Passwörtern wie in SAP Business One.

Wenn Sie sich in der App anmelden, nutzt COBI.wms zunächst diese Zugangsdaten für den COBI.wms-Login. Danach prüft die App, ob für den Service Layer ein Passwort in `apiPassword` oder `apiPass` hinterlegt ist. Sind diese Felder leer, werden einfach dieselben Zugangsdaten aus dem COBI.wms-Login verwendet, um sich beim Service Layer anzumelden.

Wenn also Benutzername und Passwort identisch mit denen des SAP-Benutzers sind, funktioniert die Anmeldung automatisch.

(Technischer Hinweis: Das Passwort eines COBI.wms-Benutzers wird nicht im Klartext, sondern als sicherer Hashwert gespeichert, da es nicht an externe Systeme übermittelt wird.)

Lizenzen importieren

Lizenzen können ganz einfach durch das Ausführen von **INSERT-Befehlen** importiert werden:

```
-- Ersetzen Sie LICENSE_1, LICENSE_2 usw. durch die tatsächlichen
Lizenzschlüssel, die Apostrophe müssen beibehalten werden.
INSERT INTO licenses (license) VALUES ('LICENSE_1');
```

```
INSERT INTO licenses (license) VALUES ('LICENSE_2');
INSERT INTO licenses (license) VALUES ('LICENSE_3');
```

Die Tabelle **licenses** enthält außerdem eine optionale Spalte **notes**, die Sie für Anmerkungen zur jeweiligen Lizenz verwenden können. Dies ist beispielsweise nützlich, wenn Sie sowohl **COBI.wms-Lizenzen** als auch **COBI.ppc-Lizenzen** in derselben Datenbank verwalten und diese voneinander unterscheiden möchten. Oder wenn Sie **Testlizenzen** importieren, die nur für einen begrenzten Zeitraum gültig sind, können Sie dies in der Notiz vermerken.

Beispiele:

```
INSERT INTO licenses (license, notes) VALUES ('LICENSE_1', 'WMS');
INSERT INTO licenses (license, notes) VALUES ('LICENSE_2', 'PPC');
INSERT INTO licenses (license, notes) VALUES ('LICENSE_3', 'PPC, gültig bis
November 2023');
```

—

Massenbearbeitung der Tabelle LICENSES

Die Information, welchem Benutzer oder Gerät eine Lizenz zugewiesen ist, befindet sich direkt in der Tabelle **LICENSES** der Management-Datenbank. Wenn Sie eine größere Anzahl von Änderungen vornehmen möchten, ist es oft am einfachsten, diese Tabelle direkt zu bearbeiten.

Beispielsweise können Sie im **MS SQL Server Management Studio** mit der rechten Maustaste auf die Tabelle **LICENSES** klicken und **„Edit Top 200 Rows“ (Oberste 200 Zeilen bearbeiten)** auswählen. Dort können Sie die Spalten **UserID** oder **DeviceID** jeder Lizenz direkt bearbeiten. *(Hinweis: Für jede Lizenz darf immer nur **eine** dieser beiden Spalten gefüllt sein – die andere muss auf **NULL** stehen.)*

Verwendung von gespeicherten Prozeduren

Zum Zuweisen von Lizenzen an Geräte oder Benutzer können Sie auch die gespeicherten Prozeduren **assignDeviceLicense** und **assignUserLicense** verwenden. Diese prüfen automatisch, ob noch freie (nicht zugewiesene) Lizenzen verfügbar sind, und verwenden eine davon:

```
-- MS SQL Server
EXEC assignDeviceLicense 1; -- Weist Gerät 1 eine freie Lizenz zu
-- oder
EXEC assignUserLicense 'user1'; -- Weist Benutzer 'user1' eine freie Lizenz
zu

-- SAP HANA
CALL assignDeviceLicense(1); -- Weist Gerät 1 eine freie Lizenz zu
-- oder
CALL assignUserLicense('user1'); -- Weist Benutzer 'user1' eine freie Lizenz
zu
```

Zum **Entziehen von Lizenzen** können Sie die Prozeduren **revokeDeviceLicense** und **revokeUserLicense** verwenden. Dadurch wird die aktuell verwendete Lizenz des entsprechenden Geräts oder Benutzers wieder freigegeben und kann anschließend einem anderen Benutzer oder Gerät zugewiesen werden:

```
-- MS SQL Server
EXEC revokeDeviceLicense 1; -- Entzieht Gerät 1 die Lizenz
-- oder
EXEC revokeUserLicense 'user1'; -- Entzieht Benutzer 'user1' die Lizenz

-- SAP HANA
CALL revokeDeviceLicense(1); -- Entzieht Gerät 1 die Lizenz
-- oder
CALL revokeUserLicense('user1'); -- Entzieht Benutzer 'user1' die Lizenz
```

Alle Module der App sind standardmäßig für **alle Geräte und Benutzer aktiviert** und müssen **manuell gesperrt** werden, wenn Sie dies ändern möchten.

Die Einstellungen für **Benutzer** haben Vorrang vor den Einstellungen für **Geräte**. Beispielsweise können Sie bestimmte Module für ein Gerät sperren – wenn sich jedoch ein Benutzer auf diesem Gerät anmeldet, der für diese Module explizit Berechtigungen erhalten hat, kann er diese Module trotzdem verwenden. Umgekehrt gilt: Wenn ein Benutzer explizit für bestimmte Module gesperrt ist, stehen diese Module für diesen Benutzer **auf keinem Gerät** zur Verfügung.

Sie können die Nutzung der App **ohne COBI.wms-Benutzerlogin effektiv blockieren**, indem Sie alle Module für alle Geräte sperren und dann die Berechtigungen gezielt an Benutzer vergeben. (Dies kann auch erreicht werden, indem in der Tabelle `Companies` die Felder `apiUsername` und `apiPassword` leer gelassen und nur für einzelne Benutzer ausgefüllt werden.)

Um Module für Geräte oder Benutzer zu aktivieren bzw. zu sperren, verwenden Sie die Prozeduren `setDevicePermission` und `setUserPermission`:

```
-- MS SQL Server
EXEC setDevicePermission 1, 'MODULE_ID', 0; -- Gerät 1 hat MODULE_ID
deaktiviert
EXEC setUserPermission 'user1', 'MODULE_ID', 1; -- Benutzer user1 hat es
aktiviert und kann es trotzdem nutzen

-- SAP HANA
CALL setDevicePermission(1, 'MODULE_ID', 0); -- Gerät 1 hat MODULE_ID
deaktiviert
CALL setUserPermission('user1', 'MODULE_ID', 1); -- Benutzer user1 hat es
aktiviert und kann es trotzdem nutzen
```

Der **erste Parameter** ist die Geräte- oder Benutzer-ID, der **zweite Parameter** ist die Modul-ID, und der **dritte Parameter** gibt den Status an: **1** bedeutet **erlaubt**, **0** bedeutet **gesperrt**.

Im obigen Beispiel ist also das Modul „MODULE_ID“ für Gerät 1 gesperrt, aber für den Benutzer „user1“ ausdrücklich freigegeben.

Nachfolgend finden Sie eine Liste der verfügbaren Modul-IDs:

* IGN: Plus-Buchung * IGE: Minus-Buchung * WTR: Umlagerung * PDN: Wareneingang * PKL: Kommissionierung * RDR: Kundenauftrag * DLN: Lieferung * RPD: Retoureneingang (Einkauf) * RDN: Rücklieferung (Verkauf) * IPE: Entnahme für Produktion * IPN: Zugang aus Produktion * PRQ: Bestellanforderung * POR: Bestellung * ITM: Artikelübersicht * INC: Inventur * WTQ: Umlagerungsanforderung * PRINT: Etikettendruck * ITEM: Artikelinfo

Zur Vereinfachung finden Sie hier eine Vorlage, um die Prozedur `setUserPermission` für **alle Module** einmal auszuführen. Sie können diesen Code direkt in SQL Server Management Studio oder HANA Studio einfügen und anpassen:

```
-- MS SQL Server
EXEC setUserPermission 'username', 'IGN', 1;
EXEC setUserPermission 'username', 'IGE', 1;
EXEC setUserPermission 'username', 'WTR', 1;
EXEC setUserPermission 'username', 'PDN', 1;
EXEC setUserPermission 'username', 'PKL', 1;
EXEC setUserPermission 'username', 'RDR', 1;
EXEC setUserPermission 'username', 'DLN', 1;
EXEC setUserPermission 'username', 'RPD', 1;
EXEC setUserPermission 'username', 'RDN', 1;
EXEC setUserPermission 'username', 'IPE', 1;
EXEC setUserPermission 'username', 'IPN', 1;
EXEC setUserPermission 'username', 'PRQ', 1;
EXEC setUserPermission 'username', 'POR', 1;
EXEC setUserPermission 'username', 'ITM', 1;
EXEC setUserPermission 'username', 'INC', 1;
EXEC setUserPermission 'username', 'WTQ', 1;
EXEC setUserPermission 'username', 'PRINT', 1;
EXEC setUserPermission 'username', 'ITEM', 1;

-- SAP HANA
CALL setUserPermission('username', 'IGN', 1);
CALL setUserPermission('username', 'IGE', 1);
CALL setUserPermission('username', 'WTR', 1);
CALL setUserPermission('username', 'PDN', 1);
CALL setUserPermission('username', 'PKL', 1);
CALL setUserPermission('username', 'RDR', 1);
CALL setUserPermission('username', 'DLN', 1);
CALL setUserPermission('username', 'RPD', 1);
CALL setUserPermission('username', 'RDN', 1);
CALL setUserPermission('username', 'IPE', 1);
CALL setUserPermission('username', 'IPN', 1);
CALL setUserPermission('username', 'PRQ', 1);
CALL setUserPermission('username', 'POR', 1);
CALL setUserPermission('username', 'ITM', 1);
CALL setUserPermission('username', 'INC', 1);
CALL setUserPermission('username', 'WTQ', 1);
CALL setUserPermission('username', 'PRINT', 1);
CALL setUserPermission('username', 'ITEM', 1);
```

Ersetzen Sie `username` durch den tatsächlichen Benutzernamen (z. B. per „Suchen und Ersetzen“ in einem Texteditor) und ändern Sie die **1** am Ende in eine **0**, um einzelne Module zu deaktivieren.

Dasselbe gilt für Geräte:

```
-- MS SQL Server
EXEC setDevicePermission deviceID, 'IGN', 1;
EXEC setDevicePermission deviceID, 'IGE', 1;
EXEC setDevicePermission deviceID, 'WTR', 1;
EXEC setDevicePermission deviceID, 'PDN', 1;
EXEC setDevicePermission deviceID, 'PKL', 1;
EXEC setDevicePermission deviceID, 'RDR', 1;
EXEC setDevicePermission deviceID, 'DLN', 1;
EXEC setDevicePermission deviceID, 'RPD', 1;
EXEC setDevicePermission deviceID, 'RDN', 1;
EXEC setDevicePermission deviceID, 'IPE', 1;
EXEC setDevicePermission deviceID, 'IPN', 1;
EXEC setDevicePermission deviceID, 'PRQ', 1;
EXEC setDevicePermission deviceID, 'POR', 1;
EXEC setDevicePermission deviceID, 'ITM', 1;
EXEC setDevicePermission deviceID, 'INC', 1;
EXEC setDevicePermission deviceID, 'WTQ', 1;
EXEC setDevicePermission deviceID, 'PRINT', 1;
EXEC setDevicePermission deviceID, 'ITEM', 1;

-- SAP HANA
CALL setDevicePermission(deviceID, 'IGN', 1);
CALL setDevicePermission(deviceID, 'IGE', 1);
CALL setDevicePermission(deviceID, 'WTR', 1);
CALL setDevicePermission(deviceID, 'PDN', 1);
CALL setDevicePermission(deviceID, 'PKL', 1);
CALL setDevicePermission(deviceID, 'RDR', 1);
CALL setDevicePermission(deviceID, 'DLN', 1);
CALL setDevicePermission(deviceID, 'RPD', 1);
CALL setDevicePermission(deviceID, 'RDN', 1);
CALL setDevicePermission(deviceID, 'IPE', 1);
CALL setDevicePermission(deviceID, 'IPN', 1);
CALL setDevicePermission(deviceID, 'PRQ', 1);
CALL setDevicePermission(deviceID, 'POR', 1);
CALL setDevicePermission(deviceID, 'ITM', 1);
CALL setDevicePermission(deviceID, 'INC', 1);
CALL setDevicePermission(deviceID, 'WTQ', 1);
CALL setDevicePermission(deviceID, 'PRINT', 1);
CALL setDevicePermission(deviceID, 'ITEM', 1);
```

Ersetzen Sie `deviceID` durch die jeweilige Geräte-ID (z. B. per Suchen und Ersetzen) und ändern Sie die **1** am Ende in eine **0**, um bestimmte Module für dieses Gerät zu deaktivieren.