

2 Import

2.1 Technische Details

Die Übertragung der Daten findet über eine CSV-Struktur statt. Dadurch können Importdateien im CSV-Format (Textdatei mit Trennzeichen Semikolon) verwendet werden.

Die benötigten Parameter für die Web-Service-Aufrufe können sowohl als http-Header-Felder oder auch als URL-Parameter mitgegeben werden.

Bei jedem Web-Service-Aufruf wird die Basic Authentication benutzt, d.h. es muss eine gültige Kombination aus Benutzername und Passwort mitgegeben werden.

Die Import-Datei wird im Message-Body übertragen. Der Inhalt der Datei wird direkt in den Message-Body geschrieben. Achtung: Multipart-Requests funktionieren nicht!

2.2 Ablauf des Imports mit Web-Service-Aufrufen

Der Import mit Web-Service-Aufrufen enthält folgende Funktionalitäten:

- Anmeldung
- Start des Imports
- Abfrage des Fortschritts
- Abholen des Protokolls
- Abbrechen des Imports
- Abmelden vom Web-Server

2.2.1 Anmeldung am Web-Server

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=Login>

Notwendige Parameter:

authorization	Basic [Base64-codierter String der Form „Benutzername:Passwort“]
---------------	--

Achtung: Der Parameter „authorization“ muss zwingend im Header übergeben werden! Der Parameter „Id“ (Tenant- oder Customer-ID) kann nur in der URL übergeben werden.

Antwort:

JSON-Objekt mit SecurityId bei erfolgreicher Anmeldung mit den Zugangsdaten

Beispiel: {"SecurityId": "40e66335-3c09-4c70-8af8-7cae5f894e95"}

2.2.2 Starten des Imports

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=Import>

Notwendige Parameter:

SecurityId	Security-ID aus den Login-Aufruf
ObjectName	[Objektname, z.B. „Mitarbeiter“]
MethodName	[Name der Importmethode, z.B. „Import Mitarbeitestammdaten Standard“]
EmailAddress (Optional)	E-Mail-Adresse, an die das Protokoll im Fehlerfall verschickt wird

(Import-Datei im Message-Body)

Antwort:

JSON-Objekt mit ImportId

Beispiel: {"ImportId": "d74bf33d-c2c7-4586-b1cf-dbcaa2b5751c"}

2.2.3 Alternative: Login und Start in einem Aufruf

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=LoginAndImport>

Notwendige Parameter:

authorization	Basic [Base64-codierter String der Form „Benutzername:Passwort“]
ObjectName	[Objektname, z.B. „Mitarbeiter“]
MethodName	[Name der Importmethode, z.B. „Import Mitarbeiterstammdaten Standard“]
EmailAddress (Optional)	E-Mail-Adresse, an die das Protokoll im Fehlerfall verschickt wird

(Import-Datei im Message-Body)

Antwort:

JSON-Objekt mit SecurityId und ImportId

Beispiel:

```
{"SecurityId": "40e66335-3c09-4c70-8af8-7cae5f894e95", "ImportId": "d74bf33d-c2c7-4586-b1cf-dbcaa2b5751c"}
```

2.2.4 Import starten und warten, bis er beendet ist

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=Import&WaitForResult=1>

Notwendige Parameter:

SecurityId	Security-ID aus dem Login-Aufruf
ObjectName	[Objektnamen, z.B. „Mitarbeiter“]
MethodName	[Name der Importmethode, z.B. „Import Mitarbeiterstammdaten Standard“]
EmailAddress (Optional)	E-Mail-Adresse, an die das Protokoll im Fehlerfall verschickt wird

(Import-Datei im Message-Body)

Antwort:

Protokolldatei als Datenstrom

2.2.5 Abfrage des Fortschritts

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=GetProgress>

Notwendige Parameter:

SecurityId	Security-ID aus dem Login-Aufruf
ImportId	Import-ID aus dem Import-Aufruf

Antwort:

JSON-Objekt mit folgenden Inhalten: ProgressMessage, ProgressValue, isFaulted, isCompleted, isCanceled

Beispiel:

```
{"IsCanceled":false,"IsCompleted":true,"IsFaulted":false,"ProgressMessage":"100%","ProgressValue":"100"}
```

2.2.6 Abholen des Protokolls

Erst verfügbar nachdem isCompleted als „true“ zurückkommt.

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=GetProtocol>

Notwendige Parameter:

SecurityId	Security-ID aus dem Login-Aufruf
ImportId	Import-ID aus dem Import-Aufruf

Antwort

Protokolldatei als Datenstrom

2.2.7 Abbrechen des Imports

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=Cancel>

Notwendige Parameter:

SecurityId	Security-ID aus dem Login-Aufruf
ImportId	Import-ID aus dem Import-Aufruf

2.2.8 Abmelden

Aufruf:

<https://de.perbit.cloud/webApi/ImportStream.ashx?Id=<Kunden-ID>&ImportAction=Logout>

Notwendige Parameter:

SecurityId	Security-ID aus dem Login-Aufruf
------------	----------------------------------

2.3 Einschränkungen und Hinweise

Es sind momentan nur Importe möglich, die nicht auf Dateien verweisen, d.h. keine Bilder oder externe Dokumente enthalten.

2.4 Fehlermeldungen

Wenn Fehler beim Starten des Imports oder vor dem Starten des Imports passieren, wird die Fehlermeldung (inkl. Stack-Trace) im Response-Body zurückgegeben.

Fehlermeldung	Ursache
Ungültige Anmeldeparameter!	Benutzername oder Passwort sind nicht korrekt.
Methode ist nicht vorhanden./Ungültiger Methodentyp.	Die angegebene Methode ist entweder nicht im System vorhanden oder es handelt sich nicht um eine Importmethode.

Methode ist nicht vorhanden.	Die angegebene Methode ist nicht im System vorhanden.
Ungültiger Methodentyp.	Die angegebene Methode ist keine Importmethode.
Um diese Aktion auszuführen sind Berechtigungen erforderlich.	Die eingestellte Rolle oder der angemeldete Benutzer (bei Rolle „0“) hat kein Öffnen-Recht auf die Importmethode.
Ungültiges Securitytoken!	Die Security-ID ist falsch oder bereits ungültig.
Tenant not found!	Die URL enthält eine falsche Customer-ID.
Es wurden keine Daten zu dieser Import-ID gefunden.	Die Import-ID gehört nicht zu einem im System vorhandenen Import oder es ist kein Protokoll zu dieser Import-ID vorhanden.
ImportAction [Wert] is invalid!	Die angegebene ImportAction existiert nicht.
Maximale Nachrichtengröße überschritten (x MB)!	Die Importdatei ist größer als das Limit, das für Importdateien konfiguriert wurde.
GUID muss 32 Ziffern mit 4 Bindestrichen enthalten (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).	Falsch aufgebaute Security-ID.
Die Sitzung wurde nicht gefunden oder ist abgelaufen!	Die Security-ID ist falsch oder bereits ungültig.
Der Wert darf nicht NULL sein. Parametername: SecurityId/ImportId	Die Import-ID bzw. Security-ID ist nicht gefüllt.

2.5 Import mit einem Import-Script

2.5.1 Benötigte Parameter für das Script

Folgende Parameter werden für das Script benötigt:

- Kunden-ID
- Pfad und Dateiname der Importdatei
- Datenobjekt, in das importiert wird
- Name der Importmethode
- Importbenutzer und Passwort

2.5.2 Aufbau des Scripts

```
#####
# Diese variablen müssen angepasst werden
#
# Ihre Kunden-ID
ScustomerID = '00000000-0000-0000-0000-000000000000'
#
# Pfad und Dateiname der Importdatei (z. B. 'D:\WebAPI\Stammdaten.txt')
SimportFile = 'D:\WebAPI\Stammdaten.txt'
#
# Objekt der Importmethode (z. B. 'Mitarbeiter')
SimportObject = 'Mitarbeiter'
#
# Name der Importmethode (z. B. 'Import Stammdaten')
SimportMethod = 'Import Mitarbeiterstammdaten Standard'
#
# Name des Importbenutzers
SimportUserName = 'API-Import-User'
#
# Passwort des Importbenutzers
SimportUserPassword = 'SICHERES_PASSWORT'
#####^#####

# Erzeugen Der URLs für die API-Funktionen
# Importbenutzer anmelden:
SurlLogin = "https://de.perbit.cloud/webApi/ImportStream.ashx?ID=$customerID&ImportAction=Login"
# Import starten:
SurlStartImport = "https://de.perbit.cloud/webApi/ImportStream.ashx?ID=$customerID&ImportAction=Import"
# Import Status abrufen:
SurlGetProgress = "https://de.perbit.cloud/webApi/ImportStream.ashx?ID=$customerID&ImportAction=GetProgress"
# Importprotokoll abrufen
SurlGetProtocol = "https://de.perbit.cloud/webApi/ImportStream.ashx?ID=$customerID&ImportAction=GetProtocol"
# Import abbrechen:
SurlCancelImport = "https://de.perbit.cloud/webApi/ImportStream.ashx?ID=$customerID&ImportAction=Cancel"
# Importbenutzer abmelden
SurlLogout = "https://de.perbit.cloud/webApi/ImportStream.ashx?ID=$customerID&ImportAction=Logout"

# Authentifizierungsstring erzeugen
Sauthorization = [convert]::ToBase64String([System.Text.Encoding]::Default.GetBytes(SimportUsername + ':' + SimportUserPassword))

# Anmelden und SecurityID ermitteln
SsecurityId = (Invoke-RestMethod -Uri SurlLogin -Method Post -Headers @{ 'authorization' = "Basic $authorization" }).SecurityId

# Import starten und ImportID ermitteln
```

```
Si mportId = (Invoke-RestMethod -Uri $urlStartImport -Method Post -Headers @{SecurityId=$securityId; ObjectName=Si mportObj ect;
Methodname=Si mportMethod} -InFile $i mportFile). ImportId

#Status des Imports überwachen
do {
    Si mportStatus = Invoke-RestMethod -Uri $urlGetProgress -Method Post -Headers @{SecurityId=$securityId; ImportId=$i mportId}
    Write-Progress -Activity "Import $i mportId" -PercentComplete $i mportStatus.ProgressValue -Status
    Si mportStatus.ProgressMessage
    Start-Sleep -Seconds 2
}
until ($i mportStatus.isCompleted -eq $true)

# Fortschrittsanzeige schließen
Write-Progress -Activity "Import $i mportId" -Completed

# Importprotokoll abrufen
Invoke-RestMethod -Uri $urlGetProtocol -Method Post -Headers @{SecurityId=$securityId; ImportId=$i mportId}

# Abmel den
Invoke-RestMethod -Uri $urlLogout -Method Post -Headers @{SecurityId=$securityId}

return

# Import abrechnen:
Invoke-RestMethod -Uri $urlCancelImport -Method Post -Headers @{SecurityId=$securityId; ImportId=$i mportId}
```

3 Automatischer Export

Es ist möglich, Exporte über den direkten Aufruf von Web-Services auszuführen. Die Schnittstelle kann per Powershell oder auch direkt aus einer anderen Anwendung (z.B. ein Lohnprogramm) heraus mit Hilfe von Web-Requests gefüllt werden.

3.1 Technische Details

Die Daten können als Text mit Trennzeichen oder Text mit fester Länge exportiert werden.

Um die Flexibilität der Schnittstelle zu erhöhen ist es möglich, die benötigten Parameter für die Web-Service-Aufrufe sowohl als http-Header-Felder als auch als URL-Parameter mitzugeben.

Achtung: Der Parameter „authorization“ muss zwingend im Header übergeben werden! Der Parameter „Id“ (Tenant- oder Customer-ID) kann nur in der URL übergeben werden.

Bei jedem Web-Service-Aufruf wird die Basic Authentication benutzt, d.h. es muss eine gültige Kombination aus Benutzername und Passwort mitgegeben werden. Wenn bereits ein Login erfolgt ist, genügt die Security-ID.

Nach erfolgreicher Anmeldung wird am WCF eine Session geöffnet und eine Security-ID zurückgegeben, deren Gültigkeit an die Session gebunden ist. Die Security-ID wird dann zur Authentifizierung in den nachfolgenden Web-Service-Aufrufen verwendet.

Die Export-Datei wird temporär in der Datenbank gehalten und verbleibt dort für einen Tag. In diesem Zeitraum kann sie mit Hilfe der Export-ID abgerufen werden. Je nach eingesetztem Web-Service-Aufruf ist u.U. kein separater Aufruf zum Abrufen der Export-Datei notwendig (s. unten)

Die Export-Datei wird im Message-Body übertragen. Der Inhalt der Datei wird direkt in den Message-Body geschrieben. Einstellungen zum Dateinamen und Speicherort in der Exportmethode werden ignoriert.

Suchkriterium und Sortierreihenfolge der Exportmethode werden angewandt. Falls das Suchkriterium Platzhalter enthält, müssen diese im entsprechenden Web-Service-Aufruf gefüllt werden.

Nach dem Schließen der Session per Timeout oder nach dem Abmelden ist die Security-ID ungültig. Durch das Abmelden wird die Session am WCF nicht sofort beendet, weil zu diesem Zeitpunkt noch Importe laufen könnten. Sie wird nach dem Ende der Importvorgänge durch das Session-Timeout beendet.

3.2 Ablauf des Exports mit Web-Service-Aufrufen

Der Export mit Web-Service-Aufrufen enthält folgende Funktionalitäten:

- Anmelden am Web-Server
- Export im Hintergrund starten
- Export starten und warten, bis er beendet ist
- Anmelden und Export starten
- Anmelden, Export starten und warten, bis er beendet ist

- Abfragen des Fortschritts
- Abbrechen eines laufenden Exports
- Exportierte Datensätze abrufen
- Abmelden

3.2.1 Anmelden am Webserver

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=Login>

Parameter	Wert
authorization	Basic [Base64-codierter String der Form „Benutzername:Passwort“]

Antwort:

JSON-Objekt mit SecurityId bei erfolgreicher Anmeldung mit den Zugangsdaten

Beispiel: {"SecurityId": "40e66335-3c09-4c70-8af8-7cae5f894e95"}

3.2.2 Export im Hintergrund starten

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=Export>

Parameter	Wert
SecurityId	Security-ID aus den Login-Aufruf
ObjectName	Objekt, in dem die Exportmethode liegt (z.B. "Mitarbeiter")
MethodName	Name der Exportmethode
ExportVariables	JSON-Objektstring für Platzhalter-Variablen der Exportmethode in der Form '[Platzhaltername]:[Wert]' z.B. '{"FirmenNr":'04','Eintrittsdatum':'01.02.2005}'

Antwort:

JSON-Objekt mit Export-ID, wenn der Export erfolgreich gestartet wurde

Beispiel: {"ExportId": "72827ed6-8950-4923-8d5b-44d8ed39f186"}

3.2.3 Export starten und warten bis er beendet ist

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=Export&WaitForResult=1>

Parameter	Wert
SecurityId	Security-ID aus den Login-Aufruf
ObjectName	Objekt, in dem die Exportmethode liegt (z.B. "Mitarbeiter")
MethodName	Name der Exportmethode
ExportVariables	JSON-Objektstring für Platzhalter-Variablen der Exportmethode in der Form '[Platzhaltername]:[Wert]' z.B. "'FirmenNr': '04', 'Eintrittsdatum': '01.02.2005'"

Antwort:

Exportierte Datensätze

3.2.4 Anmelden und Export starten

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=LoginAndExport>

Parameter	Wert
authorization	Basic [Base64-codierter String der Form „Benutzername:Passwort“]
ObjectName	Objekt, in dem die Exportmethode liegt (z.B. "Mitarbeiter")
MethodName	Name der Exportmethode
ExportVariables	JSON-Objektstring für Platzhalter-Variablen der Exportmethode in der Form '[Platzhaltername]:[Wert]' z.B. "'FirmenNr': '04', 'Eintrittsdatum': '01.02.2005'"

Antwort:

JSON-Objekt mit Security-ID und Export-ID bei erfolgreicher Anmeldung und erfolgreichem Starten des Exports

Beispiel: {"SecurityId": "83389e6c-e35a-4f85-9bdf-801eb96859ea", "ExportId": "72827ed6-8950-4923-8d5b-44d8ed39f186"}

3.2.5 Anmelden, Export starten und warten, bis er beendet ist**Aufruf:**

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=LoginAndExport&WaitForResult=1>

Parameter	Wert
authorization	Basic [Base64-codierter String der Form „Benutzername:Passwort“]
ObjectName	Objekt, in dem die Exportmethode liegt (z.B. "Mitarbeiter")
MethodName	Name der Exportmethode
ExportVariables	JSON-Objektstring für Platzhalter-Variablen der Exportmethode in der Form '[Platzhaltername]:[Wert]' z.B. "{'FirmenNr': '04', 'Eintrittsdatum': '01.02.2005'}"

Antwort:

Exportierte Datensätze

3.2.6 Abfragen des Fortschritts**Aufruf:**

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=GetProgress>

Parameter	Wert
SecurityId	Security-ID aus dem Login-Aufruf
ExportId	Export-ID aus dem Export-Aufruf

Antwort:

JSON-Objekt mit folgenden Inhalten: ProgressMessage, ProgressValue, isFaulted, isCompleted, isCanceled

Beispiel: {"IsCanceled":false,"IsCompleted":true,"IsFaulted":false,"ProgressMessage":"100%","ProgressValue":"100"}

3.2.7 Abbrechen eines laufenden Exports

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=Cancel>

Antwort:

Parameter	Wert
SecurityId	Security-ID aus dem Login-Aufruf
ExportId	Export-ID aus dem Export-Aufruf

Antwort:

Statuscode 200 ohne Inhalt

3.2.8 Exportierte Datensätze abrufen

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=GetResult>

Parameter	Wert
SecurityId	Security-ID aus dem Login-Aufruf
ExportId	Export-ID aus dem Export-Aufruf

Antwort:

Exportierte Datensätze

3.2.9 Abmelden

Aufruf:

<https://de.perbit.cloud/webApi/ExportStream.ashx?Id=<Kunden-ID>&ExportAction=Logout>

Parameter	Wert
SecurityId	Security-ID aus dem Login-Aufruf

Antwort:

Statuscode 200 ohne Inhalt

3.3 Einschränkungen und Hinweise:

Es können momentan keine externen Dokumente oder Bilder exportiert werden. Die restlichen Daten zu einem Datensatz (z.B. einem externen Dokument) werden exportiert, das externe Dokument selbst jedoch nicht.

Die Status-Flags „IsCompleted“ und „IsFaulted“ machen keine Aussage darüber, ob beim Exportieren Fehler aufgetreten sind oder nicht. Sie beziehen sich lediglich auf Fehler, die dazu geführt haben, dass der Export gar nicht durchgeführt werden konnte, wie z.B. falsche Anmeldeparameter, falsche Security-ID, ungültige Methode oder fehlende Rechte auf die Exportmethode. Fehler, die beim Ausführen des Exports auftreten werden nicht protokolliert.

Es ist keine Auswahl der Satzarten wie beim manuellen Export möglich, d.h. es werden immer alle vorhandenen Satzarten exportiert. Da automatisch ausgeführte Exporte eine Schnittstelle zu einem Drittprogramm darstellen sollten separate Exportmethoden verwendet werden, die nur die benötigten Satzarten enthalten.

3.4 Fehlermeldungen

Wenn Fehler beim Starten des Exports oder vor dem Starten des Exports passieren, wird die Fehlermeldung (inkl. Stack-Trace) im Response-Body zurückgegeben. Es wurde versucht, die gängigen Fälle abzudecken und durch hilfreiche Texte zur Ursache des Fehlers hinzuführen.

Achtung: Damit diese Fehlermeldungen im Response-Body auch beim Zugriff auf den IIS von einem anderen Rechner aus angezeigt werden ist die Einstellung " " in der web.config der Web-API notwendig. Diese Einstellung ist standardmäßig gesetzt. In der Defaultkonfiguration von IIS werden diese Meldungen nicht weitergereicht. Dann enthält der Response-Body nur die Wörter „Bad Request“.

Fehlermeldung	Mögliche Ursache
Ungültige Anmeldungsparameter!	Benutzername oder Passwort sind nicht korrekt.
Methode ist nicht vorhanden./Ungültiger Methodentyp.	Die angegebene Methode ist entweder nicht im System vorhanden oder es handelt sich nicht um eine Exportmethode.
Methode ist nicht vorhanden.	Die angegebene Methode ist nicht im System vorhanden.
Ungültiger Methodentyp.	Die angegebene Methode ist keine Exportmethode.
Um diese Aktion auszuführen sind Berechtigungen erforderlich.	Die eingestellte Rolle oder der angemeldete Benutzer (bei Rolle „0“) hat kein Öffnen-Recht auf die Exportmethode.
Ungültiges Securitytoken!	Die Security-ID ist falsch oder bereits ungültig.
Tenant not found!	Die URL enthält eine falsche Customer-ID
ExportAction [Wert] is invalid!	Die angegebene ExportAction existiert nicht.
GUID muss 32 Ziffern mit 4 Bindestrichen enthalten (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx).	Falsch aufgebaute Security-ID.
Die Sitzung wurde nicht gefunden oder ist abgelaufen!	Die Security-ID ist falsch oder bereits ungültig.
Der Wert darf nicht NULL sein. Parametername: SecurityId/ExportId	Die Export-ID bzw. Security-ID ist nicht gefüllt.
Der Wert darf nicht NULL sein. Parametername: [Platzhaltername]	Das Suchkriterium in der Exportmethode hat einen Platzhalter, der nicht gefüllt wurde.

3.5 Export mit einem Export-Script

3.5.1 Benötigte Parameter für das Script

Folgende Parameter werden für das Script benötigt:

- Kunden-ID
- Pfad und Dateiname der Exportdatei
- Datenobjekt, aus dem exportiert wird
- Name der Exportmethode
- Exportbenutzer und Passwort

3.5.2 Benötigte Rechte

Der Exportbenutzer benötigt die gleichen Daten- und Methodenrechte wie für einen "normalen" Aufruf des Exports.

3.5.3 Aufbau des Scripts

```
#####
# Diese variablen müssen angepasst werden
#
# Ihre Kunden-ID
ScustomerID = '00000000-0000-0000-0000-000000000000'
#
# Pfad und Dateiname der Exportdatei (z. B. 'D:\WebAPI\Datenexport Gesamt.txt')
SexportFile = 'D:\WebAPI\Datenexport Gesamt.txt'
#
# Objekt der Exportmethode (z. B. 'Mitarbeiter')
SexportObject = 'Mitarbeiter'
#
# Name der Exportmethode (z. B. 'Datenexport Gesamt')
SexportMethod = 'Datenexport Gesamt'
#
# Name des Exportbenutzers
SexportUserName = 'API-Export-User'
#
# Passwort des Exportbenutzers
SexportUserPassword = 'SICHERES_PASSWORT'
#####

# Erzeugen Der URLs für die API-Funktionen
# Exportbenutzer anmelden:
SurlLogin = "https://de.perbit.cloud/webApi/ExportStream.ashx?ID=$customerID&ExportAction=Login"
# Export starten:
SurlStartExport = "https://de.perbit.cloud/webApi/ExportStream.ashx?ID=$customerID&ExportAction=Export"
# Export Status abrufen:
SurlGetProgress = "https://de.perbit.cloud/webApi/ExportStream.ashx?ID=$customerID&ExportAction=GetProgress"
# Export abbrechen:
SurlCancelExport = "https://de.perbit.cloud/webApi/ExportStream.ashx?ID=$customerID&ExportAction=Cancel"
# Exportbenutzer abmelden
SurlLogout = "https://de.perbit.cloud/webApi/ExportStream.ashx?ID=$customerID&ExportAction=Logout"
# Ergebnis abrufen
SurlGetResult = "https://de.perbit.cloud/webApi/ExportStream.ashx?ID=$customerID&ExportAction=GetResult"

# Authentifizierungsstring erzeugen
Sauthorization = [convert]::ToBase64String([System.Text.Encoding]::Default.GetBytes(SexportUsername + ':' + SexportUserPassword))

# Anmelden und SecurityID ermitteln
SsecurityId = (Invoke-RestMethod -Uri SurlLogin -Method Post -Headers @{ 'authorization' = "Basic $authorization" }).SecurityId

# Export starten und ExportID ermitteln
```

```
SexportId = (Invoke-RestMethod -Uri $urlStartExport -Method Post -Headers @{SecurityId=$securityId; ObjectName=$exportObject;
Methodname=$exportMethod}). ExportId

#Status des Exports überwachen
do {
    $exportStatus = Invoke-RestMethod -Uri $urlGetProgress -Method Post -Headers @{SecurityId=$securityId; ExportId=$exportId}
    Write-Progress -Activity "Export $exportId" -PercentComplete $exportStatus.ProgressValue -Status
    $exportStatus.ProgressMessage
    Start-Sleep -Seconds 2
}
until ($exportStatus.isCompleted -eq $true)

# Fortschrittsanzeige schließen
Write-Progress -Activity "Export $exportId" -Completed

# Ergebnis abrufen
$body = (Invoke-RestMethod -Uri $urlGetResult -Method Post -Headers @{SecurityId=$securityId; ExportID=$exportId})

#Zeichensatz konvertieren
$bytes = [System.Text.Encoding]::GetEncoding("ISO-8859-1").GetBytes($body)
$body = [System.Text.Encoding]::UTF8.GetString($bytes)

# Ergebnis in Datei speichern
$body | Out-File -FilePath $exportFile

# Abmelden
Invoke-RestMethod -Uri $urlLogout -Method Post -Headers @{SecurityId=$securityId}

return

# Export abbrechen:
Invoke-RestMethod -Uri $urlCancelExport -Method Post -Headers @{SecurityId=$securityId; ExportID=$exportId}
```