

BWScheduler über REST-API steuern

Überblick

Über die REST-API können mehrere BWScheduler-Installationen verschlüsselt miteinander verbunden werden. Eine Installation übernimmt die Rolle des Servers, die übrigen Installationen arbeiten als Clients.

Eine Installation kann immer nur eine Rolle besitzen:

- **Server:** Nimmt Verbindungen an und kann Aufträge für Clients bereitstellen.
- **Client:** Verbindet sich mit dem Server, sendet Aufträge an den Server und fragt Aufträge für sich selbst ab.

Damit können Aufgaben in beide Richtungen ausgelöst werden:

- Client startet eine Aufgabe auf dem Server.
- Server startet eine Aufgabe auf einem oder mehreren Clients.

Es werden keine `SchedulerXX.RUN`-Dateien benötigt.

Voraussetzungen

- Auf allen beteiligten Systemen muss BWScheduler mindestens in Programmversion **5.31** installiert sein.
- Alle beteiligten BWScheduler müssen über das Netzwerk erreichbar sein.
- Auf dem Server muss der eingestellte TCP-Port freigegeben sein. Standard ist Port `9443`.
- Die Uhren der beteiligten Windows-Systeme sollten korrekt eingestellt sein.
- Die auszuführende Zielaufgabe muss auf der Gegenstelle vorhanden und aktiviert sein.
- Die Zielaufgabe muss über die Einstellung **REST-API Ausführung** ausdrücklich freigegeben sein.
- Zielaufgaben müssen Hauptaufgaben sein. Unteraufgaben können nicht direkt über die REST-API gestartet werden.
- Das Konto des BWSchedulers benötigt weiterhin alle Rechte, die von der Zielaufgabe benötigt werden.

Verschlüsselung und Zertifikate

Der gesamte Datenverkehr wird über TLS 1.2 verschlüsselt.

Internes Zertifikat

Wenn unter **SSL-PFX Zertifikat** keine Datei angegeben wird, erstellt und verwendet der Server automatisch ein internes Zertifikat. Dieses Zertifikat dient ausschließlich zur Verschlüsselung der Verbindung zwischen den BWSchedulern.

Für die Standardverbindung zwischen BWScheduler-Installationen muss daher kein Zertifikat bereitgestellt werden.

Eigenes PFX-Zertifikat

Optional kann auf dem Server ein eigenes PFX-Zertifikat hinterlegt werden:

- **SSL-PFX Zertifikat:** vollständiger Pfad zur PFX-Datei
- **SSL-PFX Zertifikat Kennwort:** Kennwort der PFX-Datei

Das PFX-Zertifikat muss einen privaten Schlüssel enthalten. Der im Zertifikat verwendete Name sollte zur eingestellten Serveradresse passen. Bei einem von einer vertrauenswürdigen Zertifizierungsstelle ausgestellten Zertifikat ist auf den Clients keine Ausnahme erforderlich.

Ungültige Zertifikate erlauben

Die Einstellung **Erlaube ungültige SSL-Zertifikate** betrifft die Prüfung auf der Clientseite.

- **Deaktiviert:** Zertifikat und Servername müssen gültig sein. Das automatisch verwendete interne BWScheduler-Zertifikat wird ebenfalls akzeptiert.
- **Aktiviert:** Zertifikatsfehler werden ignoriert.

“ **Sicherheitshinweis:** Aktivieren Sie diese Option nur in einem kontrollierten und vertrauenswürdigen Netzwerk.

Authentifizierung

Unter **Authentifizierungsart** stehen drei Möglichkeiten zur Verfügung:

Einstellung	Bedeutung
Keine	Für Aufgaben mit der Freigabe „Gemäß globaler Authentifizierung“ wird kein Schlüssel verlangt.
API-Key	Auf Server und Clients muss derselbe kundenspezifische API-Key eingestellt sein.

Einstellung	Bedeutung
Nur eigene Produkte	Freigegebene PHX-Produkte weisen sich automatisch mit ihrer Produktkennung und ihrem internen Produktschlüssel aus. Eine Kundeneingabe ist nicht erforderlich.

Die empfohlene Einstellung für eine Verbindung zwischen BWSchedulern ist **Nur eigene Produkte**.

Der kundenspezifische API-Key wird nur bei der Authentifizierungsart **API-Key** benötigt. Er sollte ausreichend lang und zufällig sein und darf nicht an unberechtigte Personen weitergegeben werden.

Server einrichten

Öffnen Sie die Einstellungen und wechseln Sie zum Bereich **REST-API**.

Einstellung	Wert
Aktiv	aktiviert
Betriebsart	Server
IP-Adresse des REST-Servers	kann auf dem Server leer bleiben
Port	9443 oder ein eigener freier Port
SSL-PFX Zertifikat	optional
SSL-PFX Zertifikat Kennwort	nur bei Verwendung einer PFX-Datei
Erlaube ungültige SSL-Zertifikate	auf dem Server ohne Bedeutung
Authentifizierungsart	empfohlen: Nur eigene Produkte
API-Key	nur bei Authentifizierungsart API-Key

Speichern Sie die Einstellungen und starten Sie den BWScheduler neu beziehungsweise laden Sie die Einstellungen neu.

Der Server nimmt verschlüsselte Verbindungen auf allen lokalen Netzwerkadressen über den eingestellten Port an. Stellen Sie sicher, dass die Windows-Firewall eingehende Verbindungen für diesen Port erlaubt.

Client einrichten

Öffnen Sie auf jedem Client den Bereich **REST-API**.

Einstellung	Wert
-------------	------

Aktiv	aktiviert
Betriebsart	Client
IP-Adresse des REST-Servers	IP-Adresse oder DNS-Name des Servers
Port	derselbe Port wie auf dem Server
SSL-PFX Zertifikat	normalerweise leer
SSL-PFX Zertifikat Kennwort	normalerweise leer
Erlaube ungültige SSL-Zertifikate	nur bei Bedarf aktivieren
Authentifizierungsart	dieselbe Einstellung wie am Server
API-Key	bei Authentifizierungsart API-Key derselbe Wert wie am Server

Beispiele für die Serveradresse:

```
192.168.1.20
scheduler.firma.local
https://scheduler.firma.local
```

Der Client baut ausschließlich eine ausgehende Verbindung zum Server auf. Auf dem Client muss deshalb kein eingehender REST-Port geöffnet werden.

Als Clientname wird automatisch der Windows-Rechnername verwendet. Dieser Name wird benötigt, wenn der Server einen Auftrag nur an einen bestimmten Client senden soll.

Aufgabe auf der Gegenstelle ausführen

Erstellen Sie über **Neu** die Aufgabe:

REST-API Aufgabe auf Gegenstelle ausführen

Die Aufgabe wird mit folgenden Werten vorbereitet:

Einstellung	Wert
Aufgabenart	Funktionsaufruf
App	\$RESTAPI-EXECUTE
AppParameter	Execute=22
AppParameter2	leer
Intervall	0

Passen Sie die Aufgabennummer, Ausführungszeit, Wochentage und das erlaubte Zeitfenster an den gewünschten Ablauf an.

Client startet eine Aufgabe auf dem Server

Auf dem Client wird eine Aufgabe mit folgendem AppParameter angelegt:

Execute=22

Wird diese Aufgabe auf dem Client ausgeführt, sendet er den Auftrag verschlüsselt an den Server. Der Server plant dort die aktivierte Hauptaufgabe mit der Scheduler-ID 22 ein.

Die Zielaufgabe auf dem Server muss vorhanden, aktiviert, als Hauptaufgabe eingerichtet und über **REST-API Ausführung** freigegeben sein.

REST-Freigabe der Zielaufgabe

Jede Zielaufgabe besitzt die Einstellung **REST-API Ausführung**. Standardmäßig ist eine Aufgabe nicht über REST erreichbar.

Einstellung	Verhalten
Nicht über REST ausführbar	REST-Aufrufe werden immer abgelehnt. Dies ist der Standardwert.
Gemäß globaler Authentifizierung	Es gilt die im Bereich REST-API eingestellte Authentifizierungsart.
Nur durch BWScheduler	Die Aufgabe darf ausschließlich von einem automatisch erkannten BWScheduler ausgelöst werden.
Ohne Authentifizierung	Die Aufgabe darf ohne Schlüssel ausgelöst werden. Diese Einstellung nur gezielt verwenden.

Die Freigabe wird auf der Seite eingestellt, auf der die Zielaufgabe tatsächlich ausgeführt wird. Für eine reine Sendeaufgabe mit `$RESTAPI-EXECUTE` ist keine eingehende REST-Freigabe erforderlich.

Zusammenspiel der Einstellungen

REST-API Ausführung der Zielaufgabe	Globale Einstellung	Ergebnis
-------------------------------------	---------------------	----------

Nicht über REST ausführbar	beliebig	Die Aufgabe wird immer abgelehnt.
Gemäß globaler Authentifizierung	Keine	Die Aufgabe kann ohne Schlüssel gestartet werden.
Gemäß globaler Authentifizierung	API-Key	Ein gültiger kundenspezifischer API-Key ist erforderlich.
Gemäß globaler Authentifizierung	Nur eigene Produkte	Ein gültig erkanntes PHX-Produkt ist erforderlich.
Nur durch BWScheduler	beliebig	Nur ein erkannter BWScheduler darf die Aufgabe starten.
Ohne Authentifizierung	beliebig	Die Aufgabe kann ohne Schlüssel gestartet werden.

Empfohlene Konfigurationen

BWScheduler mit BWScheduler verbinden

Auf Server und Clients:

Einstellung	Wert
Authentifizierungsart	Nur eigene Produkte
API-Key	leer

Bei der auszuführenden Zielaufgabe:

REST-API Ausführung: Nur durch BWScheduler

Diese Einstellung erfordert keine manuelle Schlüsselpflege durch den Kunden.

Externes System mit API-Key

Auf dem REST-Server:

Einstellung	Wert
Authentifizierungsart	API-Key
API-Key	langer, zufälliger und geheimer Wert

Bei der auszuführenden Zielaufgabe:

REST-API Ausführung: Gemäß globaler Authentifizierung

Das externe System muss den eingestellten API-Key bei jedem Auftrag mitsenden.

Einzelne Aufgabe ohne Authentifizierung

Bei der betreffenden Zielaufgabe:

REST-API Ausführung: Ohne Authentifizierung

Diese Freigabe gilt unabhängig von der globalen Authentifizierungsart. Verwenden Sie sie nur für gezielt ausgewählte Aufgaben und begrenzen Sie den Netzwerkzugriff zusätzlich über die Windows-Firewall.

“ **Wichtig:** Verwenden Sie für sensible Wartungs-, Programmstart- oder Beendigungsaufgaben keine anonyme Freigabe.

Server startet eine Aufgabe auf allen Clients

Auf dem Server wird eine Aufgabe mit folgendem AppParameter angelegt:

Execute=22

Der Server stellt den Auftrag allen Clients bereit, die sich seit dem Start des Servers bei ihm gemeldet haben. Jeder dieser Clients plant seine lokale Aufgabe mit der Scheduler-ID 22 ein.

Die Scheduler-ID bezeichnet immer die Aufgabe auf der jeweiligen Gegenstelle. Die Aufgaben dürfen auf Server und Clients unterschiedliche Bezeichnungen oder Inhalte besitzen.

Server startet eine Aufgabe auf einem bestimmten Client

Soll nur ein bestimmter Client angesprochen werden, wird dessen Windows-Rechnername ergänzt:

Client=PC-BUCHHALTUNG&Execute=22

Damit wird die Aufgabe 22 ausschließlich für den Client **PC-BUCHHALTUNG** bereitgestellt.

Der Clientname muss dem Windows-Rechnernamen des Zielsystems entsprechen. Groß- und Kleinschreibung sind dabei nicht relevant.

Mehrere Clients

Mehrere Clients können denselben Server verwenden. Dabei gibt es zwei Möglichkeiten:

- Nur `Execute=22`: Auftrag wird an alle beim Server bekannten Clients verteilt.
- `Client=RECHNERNAME&Execute=22`: Auftrag wird nur für den angegebenen Client bereitgestellt.

Ein Auftrag für einen bestimmten Client kann bereits bereitgestellt werden, bevor dieser wieder verbunden ist. Er wird beim nächsten erfolgreichen Abruf übernommen, solange der Server zwischenzeitlich nicht neu gestartet wurde.

Direkter REST-Aufruf am Server

Der Server nimmt einen Ausführungsauftrag unter folgender Adresse entgegen:

```
https://SERVER:9443/api/scheduler/execute
```

Ausführungsaufträge werden ausschließlich mit der HTTP-Methode `POST` angenommen. `Execute=22` wird dabei als formularcodierter Inhalt gesendet.

Bei der Authentifizierungsart **API-Key** wird der Schlüssel im HTTP-Header übergeben:

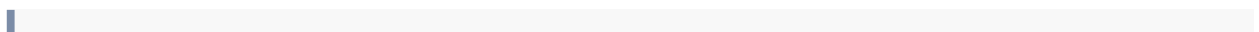
```
X-PHX-API-Key: IHR_API_KEY
```

Bei **Nur eigene Produkte** werden Produktkennung, Schlüsselkennung und Produktschlüssel automatisch durch das aufrufende PHX-Produkt übergeben. Eine manuelle Eingabe ist nicht vorgesehen.

Zur Verfügbarkeitsprüfung steht folgende Adresse bereit:

```
https://SERVER:9443/api/scheduler/status
```

Eine erfolgreiche Prüfung liefert `OK` zurück.



Sicherheitshinweis: Begrenzen Sie den Zugriff auf den REST-Port über die Windows-Firewall auf die vorgesehenen Clientsysteme beziehungsweise das interne Netzwerk.

Verhalten der Zielaufgabe

Ein angenommener REST-Auftrag verhält sich weitgehend wie ein Auftrag über eine `SchedulerXX.RUN`-Datei:

- Die Zielaufgabe wird beim nächsten Scheduler-Durchlauf eingeplant.
- Die Aufgabe muss aktiviert sein.
- Unteraufgaben können nicht direkt gestartet werden.
- Die eingestellte modale Gruppe und der Doppelstartschutz bleiben wirksam.
- Eine bereits laufende oder bereits eingeplante Aufgabe wird nicht nochmals parallel gestartet.
- Eine aktive Wartungs- oder NoLock-Sperre kann die Ausführung weiterhin zurückhalten.

Die Zeitplanung der sendenden `$RESTAPI-EXECUTE`-Aufgabe bestimmt, wann der Auftrag übertragen wird. Die feste Ausführungszeit der Zielaufgabe wird durch den angenommenen Remote-Auftrag nicht abgewartet.

Typischer Einrichtungsablauf

1. Prüfen, ob auf allen beteiligten Systemen mindestens BWScheduler-Programmversion **5.31** installiert ist.
2. Eine BWScheduler-Installation als Server konfigurieren.
3. Port 9443 oder den gewählten Port in der Server-Firewall freigeben.
4. Weitere Installationen als Clients konfigurieren.
5. Verbindung im Protokoll kontrollieren.
6. Auf der Gegenstelle eine ungefährliche Testaufgabe anlegen und aktivieren.
7. Bei der Zielaufgabe die gewünschte Einstellung unter **REST-API Ausführung** auswählen.
8. Auf der sendenden Seite eine `$RESTAPI-EXECUTE`-Aufgabe mit der entsprechenden Scheduler-ID erstellen.
9. Auftrag zunächst manuell ausführen.
10. Ausführung und Ergebnis auf beiden Seiten kontrollieren.
11. Erst danach die gewünschte Zeitplanung aktivieren.

Fehlerprüfung

Server ist nicht erreichbar

Prüfen Sie:

- REST-API ist auf beiden Seiten aktiviert.
- Server verwendet die Betriebsart **Server**.
- Client verwendet die Betriebsart **Client**.
- Serveradresse und Port stimmen überein.
- Windows-Firewall erlaubt den Port.
- Der Serverprozess läuft.

Zertifikatsfehler

Prüfen Sie:

- Stimmt der DNS-Name mit dem eigenen Zertifikat überein?
- Ist das Zertifikat noch gültig?
- Ist die Zertifikatskette auf dem Client vertrauenswürdig?
- Ist das PFX-Kennwort korrekt?

Die Option zum Ignorieren von Zertifikatsfehlern sollte nur vorübergehend zur Eingrenzung des Problems verwendet werden.

Aufgabe wird nicht ausgeführt

Prüfen Sie:

- Ist die angegebene Scheduler-ID auf der Gegenstelle vorhanden?
- Ist die Zielaufgabe aktiviert?
- Handelt es sich um eine Hauptaufgabe?
- Ist die Zielaufgabe über **REST-API Ausführung** passend freigegeben?
- Stimmen Authentifizierungsart und gegebenenfalls API-Key auf beiden Seiten überein?
- Läuft die Aufgabe bereits oder wartet sie auf ihre modale Gruppe?
- Ist eine Wartungs- oder NoLock-Sperre aktiv?
- Ist im AppParameter tatsächlich `Execute=Zahl` eingetragen?

Bestimmter Client erhält den Auftrag nicht

Prüfen Sie:

- Entspricht der Wert hinter `Client=` exakt dem Windows-Rechnernamen?
 - Hat sich der Client bereits erfolgreich beim Server gemeldet?
 - Wurde der Server nach dem Bereitstellen des Auftrags neu gestartet?
-

Revision #4

Created 2026-09-08 23:38:31 CEST by Martin Dauwa

Updated 2026-09-09 11:04:55 CEST by Martin Dauwa