



PIPE BENDING SYSTEMS | simply smart.



PIPEFAB Connect

Handbuch

V4.8.1



Änderungen:

20.11.2023, Hänsch: Dokument angelegt.
04.12.2023, Hänsch, Version 0.94, div. Ergänzungen.
01.02.2024, Hänsch, Version 1.00, Umstellung auf .NET8.
22.02.2024, Hänsch, Version 1.01, Lizenzierungsinformationen hinzugefügt.
23.02.2024, Hänsch, Version 1.02, Aktualisierung auf Hosting Pack 8.0.2.
22.05.2024, Hänsch, Version 4.4.0, Anpassung an PIPEFAB Release Version.
16.07.2024, Hänsch, Version 4.4.1, PIPEFAB WebApi appsettings.json hinzugefügt.
28.10.2024, Hänsch, Version 4.5.0, Ergänzungen zur Version 4.5.60.0.
02.12.2024, Hänsch, Version 4.5.1, Ergänzungen zum Web Api V4.5.60.2 und PIPEFAB Connect Client.
15.04.2025, Hänsch, Version 4.6.0, Ergänzungen zur Version 4.6.x, Multi-Worker-Setup.
22.04.2025, Hänsch, Version 4.6.1, Verwendung von UNC-Pfaden hinzugefügt.
21.07.2025, Hänsch: Version 4.6.2, Zusammenführung ImportWorker Scheduler restart. Delete Jobs hinzugefügt.
31.07.2025, Hänsch: Version 4.6.3, CleanUpThreshold hinzugefügt.
25.08.2025, Hänsch: Version 4.6.4, Service Berechtigungen mit Microsoft Management Console verwalten hinzugefügt.
04.09.2025, Hänsch: Version 4.6.5, Ergänzungen zur Microsoft Management Console.
18.09.2025, Hänsch: Version 4.7.0, WorkerPool-Optimierung, neue Webmethoden hinzugefügt.
12.01.2026, Hänsch: Version 4.7.1, fehlende Webmethoden hinzugefügt.
03.03.2026, Hänsch: Version 4.7.3, Review Fehlerkorrektur.
20.04.2026, Hänsch: Version 4.7.4, Update Schaubild und Fehlerkorrektur.
17.06.2026, Hänsch: Version 4.8.0, neue Funktionen für Werkstoffe hinzugefügt.
25.06.2026, Hänsch: Version 4.8.1, neue Funktionen für die Fertigungsstätte und Fertigungseinheiten hinzugefügt.



PIPE BENDING SYSTEMS GmbH & Co. KG

Hunold-Rump-Str. 76-80
57368 Lennestadt
Deutschland

Tel.: +49 2725 9540-0
Fax: +49 2725 9540-33

info@pipe-bending-systems.de
www.pipe-bending-systems.de



PIPEFAB Connect

Inhaltsverzeichnis

PIPEFAB CONNECT	0
PIPEFAB CONNECT	0
1 ALLGEMEIN.....	4
2 WAS IST PIPEFAB CONNECT.....	5
2.1 SYSTEMVORAUSSETZUNGEN	6
3 PIPEFAB CONNECT WEB-API	7
3.1 LOGIN (AUTHENTIFIZIERUNG).....	7
3.1.1 POST api/Login	7
3.1.2 POST api/Login/refresh	10
3.2 IMPORT.....	11
3.2.1 GET api/Import/job/?id=.....	12
3.2.2 POST api/Import/job	15
3.2.3 GET api/import/job/results/?id=xxxxx	17
3.2.4 GET api/import/job/logs/?id=xxxxx	19
3.2.5 GET api/import/jobs/user/.....	21
3.2.6 DELETE api/import/job.....	23
3.2.7 DELETE api/import/jobs	25
3.3 CADCONSETTINGS	28
3.3.1 GET api/cadconsettings/pcf	28
3.4 CATALOGUE.....	30
3.4.1 GET api/catalogue/list.....	30
3.4.2 GET api/catalogue/header/?id=	32
3.4.3 GET api/catalogue/find/projects/?id=	35
3.4.4 GET api/catalogue/connectortypes/?id=	37
3.4.5 GET api/catalogue/connectorcompatibilities/?id=	39
3.4.6 GET api/catalogue/parttypes/?id=.....	41
3.4.7 GET api/catalogue/partfatherandson/?id=.....	43
3.4.8 GET api/catalogue/part/?id=.....	45
3.5 CONNECTOR	49



3.5.1	GET api/connector/data/?id=	49
3.5.2	GET api/connector/type/?id=	52
3.5.3	GET api/connector/compatibility/?id=	54
3.6	MATERIAL (WERKSTOFFE)	56
3.6.1	GET api/material/byname	56
3.6.2	GET api/material/list	58
3.6.3	POST api/material/add	60
3.7	PIECEOFPIPE	64
3.7.1	POST api/pieceofpipe/job/result	64
3.8	PRODUCTION UNIT	66
3.8.1	POST api/productionunit/job/result	66
3.8.2	POST api/productionunit/job/result/drawing	68
3.9	PROJECT	70
3.9.1	GET api/project/list	70
3.10	PROJECT COMPONENTS	72
3.10.1	GET api/projectcomponent/list	72
3.11	STATE	75
3.11.1	GET api/state	75
3.12	SYMBOL	77
3.12.1	GET api/symbol/data/?id=	77
3.12.2	GET api/symbol/drawing/?id=	79
3.13	TECHNOLOGY CONFIG DATA (LOOKUP-DATENTYPEN)	81
3.13.1	GET api/technologyconfig/data/partconnectorassemblytype	81
3.13.2	GET api/technologyconfig/data/substitutiontype	83
3.13.3	GET api/technologyconfig/data/weldconnectiontype	84
3.14	WORK LOCATION (FERTIGUNGSSTÄTTE)	87
3.14.1	GET api/worklocation/list	87
3.15	WORK PACKAGE (FERTIGUNGSEINHEIT)	89
3.15.1	POST api/workpackage/isometricProcessingStatus	89
4	PIPEFAB CONNECT CLIENT (NUGET-PACKAGE)	93
4.1	BEISPIEL GETSTATE	94

4.2	BEISPIEL IMPORT PCF	95
5	INSTALLATION	96
5.1	PIPEFAB CONNECT WEB-API	96
5.1.1	Systemvoraussetzungen.....	96
5.1.2	Lizensierung.....	97
5.1.3	Web Server IIS installieren	97
5.1.4	ASP.NET Core Hosting Bundle für .NET 9 installieren	102
5.1.5	Anzeige der installierten .NET Versionen.....	103
5.1.6	PIPEFAB Connect Verzeichnis anlegen	104
5.1.7	Web Server IIS für PIPEFAB Connect WebApi einrichten	105
5.1.8	Programmdateien kopieren	109
5.1.9	Datenbankzugriff und Lizenzserver in appsettings.json einstellen.....	109
5.1.10	HTTPS-Zugriff einrichten	110
5.1.11	Funktionstest.....	111
5.2	PIPEFAB CONNECT IMPORTWORKER SERVICE.....	112
5.2.1	Systemvoraussetzungen.....	112
5.2.2	Lizensierung.....	112
5.2.3	PIPEFAB Connect Verzeichnis anlegen	113
5.2.4	Programmdateien kopieren	113
5.2.5	Datenbankverbindung konfigurieren.....	113
5.2.6	Verbindung zum Lizenzserver konfigurieren	114
5.2.7	PIPEFAB Connect ImportWorker konfigurieren	115
5.2.8	Konfiguration des Worker-Pool's	117
5.2.9	PIPEFAB Connect ImportWorker Service installieren	119
5.2.10	Funktionstest.....	140
5.2.11	PIPEFAB Connect ImportWorker-Service clean up	140
5.2.12	Zugriff auf Netzwerklaufwerke (UNC-Pfad)	153
6	BEISPIELKONFIGURATIONEN PIPEFAB CONNECT IMPORTWORKER	155
6.1	EIN IMPORTWORKER.....	155
6.2	MEHRERE IMPORTWORKER	156



6.3	ZWEI WORKER-POOL'S MIT JE ZWEI IMPORTWORKERN.....	157
6.4	EIN WORKER-POOL MIT DREI IMPORTWORKERN	159

1 Allgemein

Dieses Handbuch gilt die die PIPEFAB Connect Versionen 4.6.x.x.

Die Bedienungsgrundlagen beschreiben den allgemeinen Umgang mit PIPEFAB 4 unter Microsoft Windows™ 10 und Windows™ 11.

Die Produktnamen sind von der PIPE BENDING SYSTEM GmbH & Co. KG geschützt.

Urheberschutzrechte gelten ebenfalls für die Produkte Oracle und Microsoft Windows. Die vorliegende Dokumentation bezieht sich auf die Oracle-Version von PIPEFAB 4.

Bitte beachten Sie, dass für PIPEFAB 4 unterschiedliche Programmpakete und Optionen existieren. Gegebenenfalls sind nicht alle in diesem Handbuch beschriebenen Funktionen in Ihrem Softwareumfang enthalten. Sollte ausschließlich eine Option beschrieben sein, finden Sie Informationen zu den anderen Programmteilen in separaten Handbüchern.

Wir behalten uns das Recht vor, Änderungen in der Software und in diesem Handbuch ohne besondere Erwähnung vorzunehmen.

Hinweis: Alle Abbildungen und Daten (z.B. Parameter der Rohre, Parameter der Maschinen und Werkzeuge, Isometrien und Biegedaten) in diesem Handbuch basieren auf Testdaten und stellen keine wirklichen Konstruktionen dar. Sie dienen nur zu Dokumentationszwecken.

2 Was ist PIPEFAB Connect

PIPEFAB Connect stellt ein Web-API's und die für den Import von CAD-Daten notwendigen sog. ImportWorker Service zur Verfügung. Über das PIPEFAB Connect Web-API kann auf einen Teil des PIPEFAB Funktionsumfangs zugegriffen werden.

Ein Web-API (Application Programming Interface) ist eine Schnittstelle, die es verschiedenen Anwendungen ermöglicht, miteinander zu kommunizieren und Daten auszutauschen. Sie wird verwendet, um die Funktionalität von PIPEFAB anderen Entwicklern zur Verfügung zu stellen, damit diese darauf zugreifen und sie in ihre eigenen Anwendungen integrieren können.

Das PIPEFAB Connect Web-API ermöglicht es Entwicklern, Anfragen an eine bestimmte URL zu senden und Daten im JSON-Format zu empfangen.

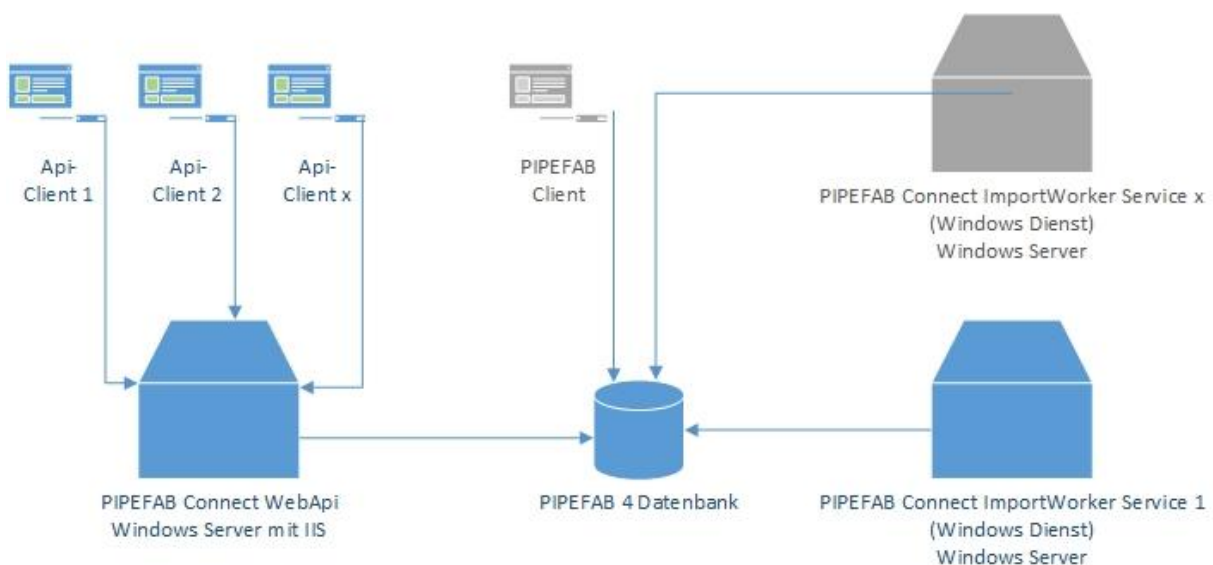
Technische Details und der Funktionsumfang des PIPEFAB Connect Web-API's, können den folgenden Kapitel entnommen werden.

2.1 Systemvoraussetzungen

PIEFAB Connect lässt sich in eine bestehende PIEFAB Systemumgebung wie folgt integrieren.

- Für die Bereitstellung des PIEFAB Connect Web-API's wird ein aktueller Windows Server mit installierten IIS (Internet Information Server) benötigt. Auf diesem Windows Server/IIS muss das ASP.NET Core Hostingpaket für .NET 9 installiert sein. Für den verschlüsselten Zugriff mittel HTTPS wird zudem ein entsprechendes SSL-Zertifikat für den IIS benötigt. Der Zugriff auf die vorhandene PIEFAB 4 Datenbank muss gewährleistet sein.
- Für die Verarbeitung der über das PIEFAB Connect WebApi's empfangenden Import-Daten wird der PIEFAB Connect ImportWorker Service benötigt. Der PIEFAB Connect ImportWorker Service läuft als Windows Dienst und sorgt für die Verarbeitung der zu importierenden Daten. Hierfür wird ein aktueller Windows Server benötigt. Der Zugriff auf die vorhandene PIEFAB 4 Datenbank muss gewährleistet sein.

Die folgende Abbildung zeigt schematisch den Systemaufbau von PIEFAB Connect. Ein oder mehrere Clients kommunizieren mit dem PIEFAB Connect WebApi. Das PIEFAB Connect WebApi speichert die Daten in der zentralen PIEFAB Datenbank. Der oder die PIEFAB ImportWorker Services verarbeiten die Daten aus der PIEFAB Datenbank.



Api-Client => PIEFAB ImportManager, PIEFAB ControlPanel oder Ihre Applikation

PIEFAB Client => PIEFAB 4 Windows Applikation

Weitere Details zum Installationsvorgang stehen im Kapitel 4 zur Verfügung.

3 PIPEFAB Connect Web-API

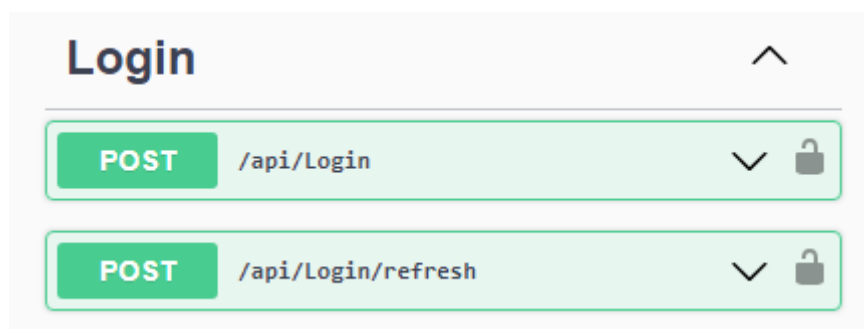
Das PIPEFAB Connect Web-API stellt in der aktuellen Version die im folgenden beschriebenen Zugriffspunkte zur Verfügung.

Der Zugriff auf das Web-API ist nur über eine vorgelagerte Authentifizierung möglich.

3.1 Login (Authentifizierung)

Der Zugriff auf das Web-API erfordert eine Authentifizierung des Anwenders der auf das Web-API zugreifen möchte.

Die folgende Abbildung zeigt die zur Verfügung stehenden Login-Zugriffspunkte.



3.1.1 POST api/Login

Über den Webmethode *api/Login* meldet sich der Anwender am PIPEFAB Connect Web-API an und erhält im Erfolgsfall ein sog. JWT (Erläuterung siehe unten) welches bei jedem folgenden Zugriff auf das Web-API mit dem Request an das Web-API übergeben werden muss. Das JWT besitzt eine festgelegte Gültigkeitsdauer

JWT steht für JSON Web Token. Es handelt sich hierbei um ein offenes Standardformat zur sicheren Übertragung von Informationen. JWT's werden zur Authentifizierung und Autorisierung in Webanwendungen verwendet.

POST

/api/Login
PIPEFAB Connect login with username and password to get a new PIPEFAB Connect security token and a refresh token.




Parameters

No parameters

Request body

application/json

LoginRequest: username and password.

Example Value

Schema

```
{
  "username": "string",
  "password": "string"
}
```

Responses

Code	Description	Links
200	OK Media type application/json Controls Accept header. Example Value Schema <pre>{ "token": "string", "username": "string", "expiration": "2024-10-29T08:04:07.627Z", "refreshTokenExpiryDate": "2024-10-29T08:04:07.627Z", "refreshToken": "string", "createDate": "2024-10-29T08:04:07.627Z" }</pre>	No links
400	Bad Request	No links

Beispiel:

Request: <https://pipefabconnect/api/login>

Im Request-Body bitte den Usernamen und das Passwort übergeben.

```
{
  "Username": "MeinUser",
  "Password": "MeinPasswort"
}
```

Response:

```
{  
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 eyJldGkiOiJEZTlzdkhqcS90aF3ejdNeT  
JBOEN3PT0iLCJlc2VyIjoic3lzdgVtIiwidW5psvVLX25hbWUiOiJTTeXN0ZW0gTWFWYWNldciIsIlByb2Zpb  
CI6IkFETUl0IiwiaGFuZC5kaXI6Imh0dHA6Ly9yY2hlbnWFzfLnhtbHNvYXAub3JnL3dzLzlwMD  
UwMDeWUvaWRlbmRpdHkvY2xhaW1zLzE2fDFlZGlhbnRpY2FOaW9uIjoidG9rZW4iLCJuYmYiOiJlcyE3MzMzMTE3MmcnR  
mV4cCI6MTczMDExNzcwOCwiaXNzIjoibG9rZW4iLCJuYmYiOiJlcyE3MzMzMTE3MmcnRmFi  
LkNvbml5YQ3QuVXNlciJ9 il0rlUmRkoJCblTmtPzdHQKzx9qXCJBjqTXdB-BODE",  
    "username": "MeinUser",  
    "expiration": "2024-10-28T13:15:08.0304501+01:00",  
    "refreshTokenExpiryDate": "2024-10-28T14:15:08.0305756+01:00",  
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 eyJodHRwOi8vc2NoZW1hcyc54ert  
bWxzZ2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcsqy9hdXRoZW50aWNhndGlvdviI6InJlZnJlc  
2giLCJVjc2VyIjoic3lzdgVtIiwibmJmIjoixNzMWMTExNzA4LCJleHAiOiJlcyE3MzMzMTE3MmcnRmDsImlzcyl6IiBp  
cGVGYWYuQ29ubmVjdC5CYWNrZW5kIiwiaXYvbkIjoibG9rZW4iLCJuYmYiOiJlcyE3MzMzMTE3MmcnRmFsImFya  
5PPHCWZCI7eHhoWhflKdxE8bJAD4pEc",  
    "createDate": "2024-10-28T11:35:08.0503747+01:00"  
}
```

Im Response ist sowohl das Sicherheitstoken `[token]` als auch das Refresh-Token `[refreshToken]` enthalten. Das Sicherheitstoken muss bei jedem Aufruf von gesicherten PIPEFAB Connect Webmethoden verwendet werden. Das Sicherheitstoken besitzt eine Gültigkeitsdauer. Siehe Feld `[expiration]`.

3.1.2 POST api/Login/refresh

Das PIPEFAB Connect Sicherheitstoken muss in regelmäßigen Abständen erneuert werden. Um dieses ohne erneute Anmeldung an das PIPEFAB Connect Web-API durchzuführen, muss das beim Login ausgestellte Refresh-Token verwendet werden. Das Refresh-Token besitzt eine längere Gültigkeitsdauer (ca. 1 Stunde) als das PIPEFAB Connect Sicherheitstoken. Nach dem Ablauf des Refresh-Tokens muss eine erneute Anmeldung über die in 3.1.1 beschriebene Methode erfolgen.

POST

/api/Login/refresh
Refreshes a valid PIPEFAB Connect security token and returns a new security token and a new refresh token.

🔒 ^

Parameters

No parameters

Responses

Code	Description	Links
200	OK Media type application/json Controls Accept header. Example Value Schema <pre>{ "token": "string", "username": "string", "expiration": "2024-10-29T08:08:04.912Z", "refreshTokenExpiryDate": "2024-10-29T08:08:04.912Z", "refreshToken": "string", "createDate": "2024-10-29T08:08:04.912Z" }</pre>	No links
401	Unauthorized	No links

3.2 Import

Um den Import eines PCF-Files über den PIPEFAB Connect ImportWorker Service zu starten, muss im ersten Schritt ein Import-Auftrag über das PIPEFAB Connect WebApi angelegt werden. Der PIPEFAB Connect ImportWorker Service prüft periodisch ob neue Importaufträge vorhanden sind und verarbeitet ggf. vorhandene Importdaten.

Die folgende Abbildung zeigt die zur Verfügung stehenden Import-Zugriffspunkte.

Import ^		
GET	/api/Import/job	Get an import job by jobld. 🔒 ✓
POST	/api/Import/job	Import an CAD file to PIPEFAB Connect 🔒 ✓
GET	/api/Import/job/logs	Get log entries for a job bei jobld. 🔒 ✓
GET	/api/Import/job/results	Get job results by jobld. 🔒 ✓
GET	/api/Import/jobs/user	Get all jobs for the current security token user. 🔒 ✓
GET	/api/Import/cadconsettings	Get the PIPEFAB CadCon settings for pcf files. 🔒 ✓

3.2.1 GET api/Import/job/?id=....

Gibt zu der angegebenen JobId (Parameter id) Informationen über den aktuellen Bearbeitungszustand des zugehörigen Import-Jobs zurück.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Import/job

Get an import job by jobld.

🔒 ^

Parameters

Try it out

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type

application/json

▼

Code	Description
200	OK
	Example Value Model <pre>{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "state": "string", "type": "string", "interface": "string", "creationDate": "2025-04-24T09:40:18.965Z", "params": "string", "user": "string", "workerPool": "string", "workerNumber": 0, "workerHost": "string", "importMode": 0, "fullName": "string", "osCreator": "string", "fileName": "string" }</pre>
401	Unauthorized
404	Not Found

Beispiel:

Abfrage des aktuellen Bearbeitungszustandes für den Import-Job mit der Id = 2ccedf60-b949-4b5b-acc1-8fcbc410122b

Request:

<https://pipefabconnect/api/Import/Job/?id=2ccedf60-b949-4b5b-acc1-8fcbc410122b>

Response:

```
{
  "id": "2ccedf60-b949-4b5b-acc1-8fcbc410122b ",
  "state": "Finished",
  "type": "CADImportCheckAndSave",
  "interface": "PCF",
  "creationDate": "2024-10-30T09:26:05.08",
  "params": "PCF-Import",
  "user": "TestUser",
  "workerPool": "",
  "workerNumber": 1,
  "workerHost": "pipefabconnect",
  "fileName ": "GetDrawingFrameNameByQuantity-19_edit"
}
```

Parameter	Bezeichnung
id	Die zugehörige JobId.
state	Der Bearbeitungszustand. Mögliche Werte: - Waiting => Wartet auf die Verarbeitung - QueuedWx => Reserviert für die Verarbeitung mit Worker x - Running => Verarbeitung läuft gerade - Finished => Verarbeitung abgeschlossen
type	Importtyp. Mögliche Werte: - CADImportCheckOnly => CAD-Import nur Prüfen ohne Speicherung - CADImportCheckAndSave => CAD-Import inkl. Prüfung und Speicherung
interface	Das zu verwendete Importinterface. Mögliche Werte: - PCF => Importiert eine Datei im PCF-Format
creationDate	Datum und Uhrzeit an dem der Job/Auftrag erzeugt wurde.
params	Name der PIPEFAB CadCon-Einstellungen die für den Import verwendet wurden/werden.
user	Der für den Import zuständige PIPEFAB Benutzer.

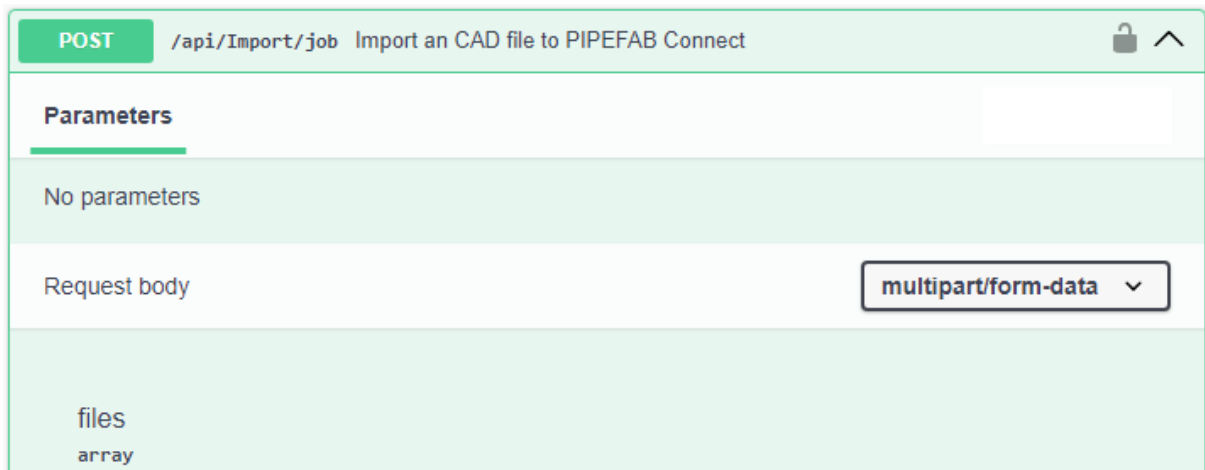


Parameter	Bezeichnung
workerPool	Name des WorkerPools dem dieser Job/Auftrag für die Verarbeitung zugeordnet wird. Der Parameter „WorkerPool“ ist ein optionaler Parameter und kann, wenn er nicht benötigt wird, weggelassen werden.
workerNumber	Die Nummer des Workers der diesen Auftrag/Job verarbeitet hat.
workerHost	Der Hostname (Servername) auf dem der Worker installiert ist
filename	Der Dateiname der importierten CAD Datei ohne Extension.

3.2.2 POST api/Import/job

Erzeugt einen neuen Import-Job zum Importieren einer CAD-Datei und übergibt die Daten an PIPEFAB Connect. Liefert eine eindeutige JobId (Guid) zurück.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.



POST /api/Import/job Import an CAD file to PIPEFAB Connect

Parameters

No parameters

Request body multipart/form-data

files
array

Request-Daten:

Multipart/Form-Data:

file = [PCF-File]

interface = PCF

params = [CadConSettingName]

type = CADImport

ImportMode = CheckAndSave

WorkerPool = TestPool

Parameter	Bezeichnung
file	Die zu importierende CAD-Datei
interface	Das zu verwendete Importinterface. Mögliche Werte sind: PCF => Importiert eine Datei im PCF-Format
params	Name der PIPEFAB CadCon-Einstellungen die für den Import verwendet werden sollen.
type	Der zu verwendete Importtyp: Mögliche Werte sind: CADImport => Importiert eine CAD-Datei
importMode	Der zu verwendete Import Modus. Mögliche Werte sind: CheckOnly => Prüfen der Daten ohne Speicherung. CheckAndSave => Prüfen der Daten und wenn möglich Speichern.

Parameter	Bezeichnung
WorkerPool	Name des WorkerPools dem dieser Job/Auftrag für die Verarbeitung zugeordnet wird. Der Parameter „WorkerPool“ ist ein optionaler Parameter und kann, wenn er nicht benötigt wird, weggelassen werden.

Beispiel:

☐ none
 ☒ form-data
 ☐ x-www-form-urlencoded
 ☐ raw
 ☐ binary
 ☐ GraphQL

	Key		Value
☒	file	File ▾	 MultiWorkerTestFile1.pcf 
☒	interface	▾	PCF
☒	params	▾	PBS_User_PCF_Import
☒	type	Text ▾	CADImport
☒	ImportMode	▾	CheckAndSave
☒	WorkerPool	Text ▾	TestPool

Responses		
Code	Description	Links
200	OK Media type application/json ▾ Controls Accept header. Example Value Schema <pre>{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" }</pre>	No links
400	Bad Request	No links
413	Content Too Large	No links
500	Internal Server Error	No links

3.2.3 GET api/import/job/results/?id=xxxxx

Gibt zu der angegebenen JobId (Parameter id) das Ergebnis des Imports zurück.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Import/job/results

Get job results by jobId.

🔒 ⬆

Parameters

Name	Description
id string(\$uuid) (query)	id

Responses

Code	Description	Links
200	OK Media type text/plain Controls Accept header. Example Value Schema <pre>[{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "payload": "string", "message": "string", "succeeded": true }]</pre>	No links
401	Unauthorized	No links
404	Not Found	No links

Inhalt des Response

Response	Bezeichnung
jobId	Die zugeordnete JobId.

Response	Bezeichnung
payload	Das Ergebnis des Imports. Im Erfolgsfall wird im Payload die drawingId mitgeteilt.
message	Meldungen zum Job-Import.
succeeded	Mögliche Werte sind: True = Erfolgreich, False = Nicht erfolgreich, Details können über das api/siehe Job-Results Abgefragt werden.

Beispiel:

In diesem Beispiel werden die Ergebnisse des zuvor importierten Jobs mit der Id = "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae" abgefragt.

Request:

<https://pipefabConnect/api/import/job/results/?id=9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae>

Response:

```
{
  "jobid": "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae",
  "payload": " drawingId=b6f0d019-aaee-4946-9347-6f11d23765f0;",
  "message": "",
  "succeeded": true
}
```

Im Response dieses Beispiel wird der Wert „succeeded = true“ mitgeteilt. Das bedeutet, die Bearbeitung des Jobs konnte mit PIPEFAB Connect erfolgreich durchgeführt werden. Im Payload steht die von PIPEFAB generierte drawingId. Mit Hilfe der drawingid und der zugehörigen JobId kann auf weitere Daten wie z.B. die PieceOfPipe-Daten zugegriffen werden.

3.2.4 GET api/import/job/logs/?id=xxxxx

Gibt zu der angegebenen JobId (Parameter id) die vorhandenen Log-Meldungen des Imports zurück.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Import/job/logs

Get log entries for a job bei jobld.

🔒 ⬆

Parameters

Name	Description
id	id
string(\$uuid)	
(query)	

Responses

Code	Description	Links
200	OK Media type text/plain Controls Accept header. Example Value Schema <pre>[{ "text": "string", "severity": "string", "entryNr": 0, "logTime": "2024-11-07T08:54:55.237Z" }]</pre>	No links
401	Unauthorized	No links
404	Not Found	No links

Inhalt des Response

Eine Liste von Log-Meldungen mit folgendem Inhalt.

Response	Bezeichnung
text	Meldungstext
severity	I = Information W = Warnung E = Fehler S = Erfolgreich
entryNr	Nummer der Meldung.
logTime	Zeitstempel der Meldung.

Beispiel:

In diesem Beispiel werden die Log-Meldungen des zuvor importierten Jobs mit der Id = "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae" abgefragt.

Request:

<https://pipefabConnect/api/import/job/logs/?id=9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae>

Response:

```
[
  {
    "text": "Die Länge des Rohres wurde für die Schweißnahtbreiten geändert.",
    "severity": "I",
    "entryNr": 1,
    "logTime": "2024-10-30T09:26:18.857"
  },
  {
    "text": "Seite B wird zuerst gebogen.",
    "severity": "I",
    "entryNr": 2,
    "logTime": "2024-10-30T09:26:20.287"
  },
  .....
  .....
]
```

3.2.5 GET api/import/jobs/user/

Gibt zu der angegebenen JobId (Parameter id) Informationen über den aktuellen Bearbeitungszustand des zugehörigen Import-Jobs zurück.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Import/jobs/user

Gibt alle Jobs für den aktuell angemeldeten User (security token user) zurück.

03.04.2025, Hänsch: Parameter maxJobCount hinzugefügt.

Parameters

Name	Description
maxJobCount integer(\$int32) (query)	maxJobCount

Responses

Response content type

application/json

Code	Description
200	OK Example Value Model <pre>[{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "state": "string", "type": "string", "interface": "string", "creationDate": "2026-01-12T15:06:44.699Z", "finishedDate": "2026-01-12T15:06:44.699Z", "params": "string", "user": "string", "workerPool": "string", "workerNumber": 0, "workerHost": "string", "fileName": "string" }]</pre>
400	Bad Request
401	Unauthorized

Beispiel:

Abfrage des aktuellen Bearbeitungszustandes der Import-Jobs, die dem aktuellen Login-User zugeordnete sind. Mit dem optionalen Parameter maxJobCount kann die Anzahl der zurückgelieferten Jobs beschränkt werden. In diesem Beispiel werden nur die ersten 10 Ergebnisse benötigt. Demzufolge ist maxJobCount = 10 gesetzt. Wird der Parameter maxJobCount nicht angegeben, werden alle Jobs zu dem aktuellen Login-user zurückgeliefert.

Request:

<https://pipefabconnect/api/Import/Jobs/user/?maxjobcount=10>

Response:

```
{
  "id": "047830c6-bc53-402a-85b5-db59d4eb5f06",
  "state": "Finished",
  "type": "CADImportCheckAndSave",
  "interface": "PCF",
  "creationDate": "2025-11-17T09:56:41.06",
  "finishedDate": "2025-11-17T09:57:24.93",
  "params": {
    "PBS_User_PCF_Import": {
      "user": "PBS_User",
      "workerPool": "",
      "workerNumber": 1,
      "workerHost": "pbs-vwd-kt01",
      "fileName": "MultiWorkerTestFile10"
    }
  },
  ...
}
```

3.2.6 DELETE api/import/job

Löscht einen importierten Job. Der zu löschende Job muss sich im Status „finished“ befinden. Gelöscht werden der Job und die zugehörenden Log- und Result-Einträge. Die aus dem Import-Job resultierende Isometrie und die zugehörigen Fertigungsdaten werden aus Revisionsgründen nicht gelöscht und stehen im PIPEFAB Explorer weiterhin zur Verfügung.

Achtung: Wenn ein Job gelöscht wurde, kann über das PIPEFAB Connect WebApi nicht mehr auf die zum Job erzeugten Daten wie z.B. die PieceOfPipe- oder ProductionUnits-Daten zugegriffen werden.

Löscht den Job mit der angegebenen JobId (Parameter id).

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

DELETE

/api/Import/job

Löscht einen Job

Parameters

Try it out

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type

application/json

Code	Description
200	OK
	Example Value Model
	<pre>{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "isDeleted": true, "message": "string", "messageId": 0 }</pre>
401	Unauthorized

Inhalt des Response

Response	Bezeichnung
jobId	Die zugeordnete JobId.
isDeleted	Das Ergebnis des Imports. Im Erfolgsfall wird im Payload die drawingId mitgeteilt.
message	Meldungen zum Job-Import.
messageId	Mögliche Werte sind: True = Erfolgreich, False = Nicht erfolgreich, Details können über das api/siehe Job-Results Abgefragt werden.

Beispiel:

In diesem Beispiel wird der Job mit der Id = "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae" gelöscht.

Delete Request:

<https://pipefabConnect/api/import/job/?id=9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae>

Response:

```
{
  "jobId": "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae",
  "isDeleted": true,
  "message": "",
  "messageId": 0
}
```

Folgende Fehlermeldungen können auftreten:

Message Id	Meldung
0	Keine Meldung vorhanden. Alles ok.
1000	Die angegebene JobId existiert nicht.
1001	Zu löschender Job befindet sich gerade in Bearbeitung und kann deshalb nicht gelöscht werden.
1002	Zu löschender Job befindet sich gerade in der Warteschlange für den Import und kann deshalb nicht gelöscht werden.
1003	Beim Löschen des Jobs ist ein unerwarteter Fehler aufgetreten.

3.2.7 DELETE api/import/jobs

Löscht eine Liste von importierten Jobs. Die zu löschenden Jobs müssen sich im Status „finished“ befinden. Gelöscht werden die Jobs und die zugehörenden Log- und Result-Einträge. Die aus dem jeweiligen Import-Job resultierende Isometrie und die zugehörigen Fertigungsdaten werden aus Revisionsgründen nicht gelöscht und stehen im PIPEFAB Explorer weiterhin zur Verfügung.

Achtung: Wenn ein Job gelöscht wurde, kann über das PIPEFAB Connect WebApi nicht mehr auf die zum Job erzeugten Daten wie z.B. die PieceOfPipe- oder ProductionUnits-Daten zugegriffen werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

DELETE /api/Import/jobs Löscht alle Job's deren Id's in der Liste übergeben wurden.

Parameters
Try it out

Name	Description
body object (body)	Example Value Model <pre>{ "jobIdList": ["3fa85f64-5717-4562-b3fc-2c963f66afa6"] }</pre> Parameter content type application/json

Responses

Response content type application/json

Code	Description
200	OK Example Value Model <pre>{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "isDeleted": true, "message": "string", "messageId": 0 }</pre>
400	Bad Request
401	Unauthorized

Inhalt des Response

Eine Liste von Objekten zu jeder JobId die im Request übergeben worden ist. Zu jeder übergebenen JobId enthält der Response die folgenden Felder:

Response	Bezeichnung
jobId	Die zugeordnete JobId.
isDeleted	Das Ergebnis des Imports. Im Erfolgsfall wird im Payload die drawingId mitgeteilt.
message	Meldungen zum Job-Import.
messageld	Mögliche Werte sind: True = Erfolgreich, False = Nicht erfolgreich, Details können über das api/siehe Job-Results Abgefragt werden.

Beispiel:

In diesem Beispiel werden zwei Jobs gelöscht.

JobId 1 = 8637a8c3-19d1-4e2c-8cd4-15855e2abe04

JobId 2 = 1a8a3ed0-21ea-42ec-bac5-c87971aa0242

JobId 3 = 3ff28860-4bcc-4321-998b-1f725ab5717a

Request:

<https://pipefabconnect/api/import/jobs>

Request-Body:

```
{
  "jobIdList": [
    "8637a8c3-19d1-4e2c-8cd4-15855e2abe04",
    "1a8a3ed0-21ea-42ec-bac5-c87971aa0242",
    "3ff28860-4bcc-4321-998b-1f725ab5717a"
  ]
}
```

Response:

```
[
  {
    "jobId": "8637a8c3-19d1-4e2c-8cd4-15855e2abe04",
    "isDeleted": true,
    "message": "",
    "messageId": 0
  },
  {
    "jobId": "1a8a3ed0-21ea-42ec-bac5-c87971aa0242",
    "isDeleted": true,
    "message": "",
    "messageId": 0
  },
  {
    "jobId": "3ff28860-4bcc-4321-998b-1f725ab5717a",
    "isDeleted": true,
    "message": "",
    "messageId": 0
  }
]
```

3.3 CadConSettings

3.3.1 GET api/cadconsettings/pcf

Mit Hilfe dieser Webmethode können die im PIPEFAB gespeicherten CadCon-Einstellungen gelesen werden. Zurückgegeben werden lediglich die Namen unter der die CadCon-Einstellungen im PIPEFAB gespeichert wurden. Bei einem Import einer PCF-Datei muss der Name der zu verwendeten CadCon-Einstellungen mit angegeben werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/CadConSettings/pcf

Get the available CadCon pcf setting names from PIPEFAB to use it with an import job.

🔒 ⬆

Parameters

No parameters

Responses

Code	Description	Links
200	OK Media type application/json Controls Accept header. Example Value Schema <pre>{ "username": "string", "importType": "string", "settingsName": "string", "userInit": "string" }</pre>	No links
401	Unauthorized	No links

Beispiel:

In diesem Beispiel werden die aktuell in PIPEFAB vorhandenen Namen der CadCon-Einstellungen für einen möglich PCF-Import gelesen.

Request:

<https://pipefabconnect/api/import/cadconsettings/pcf>

Response:

```
[
  {
    "username": "User",
    "importType": "PCFIMPORT",
    "settingsName": "PCF-Einstellungen01",
    "userInit": "usr2034",
  },
  {
    "username": "User",
    "importType": "PCFIMPORT",
    "settingsName": "PCF-Einstellungen02",
    "userInit": "system",
  }
]
```

Es wurden zwei gespeicherte CadCon-Einstellungssätze für den PCF-Import gefunden. Beim Import einer PCF-Datei muss der Name der zum Import zu verwendeten Einstellungen angegeben werden. In diesem Fall z.B. `PCF-Einstellungen02`.

3.4 Catalogue

3.4.1 GET api/catalogue/list

Mit Hilfe dieser Webmethode kann eine Liste der in PIPEFAB vorhandenen Kataloge abgerufen werden. Dabei werden nur Kataloge berücksichtigt die auch in Projekten verwendet werden dürfen.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/list

Get a list of all open PIPEFAB catalogue names

🔒 ⬆

Parameters

No parameters

Responses

Response content type

application/json

Code	Description
200	OK
	Example Value Model <pre> { "catalogues": [{ "catalogueName": "string", "versionNumber": 0, "versionText": "string", "description": "string", "modifiedBy": "string", "modifiedDate": "2025-12-11T15:16:28.247Z", "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "dimensionUnits": "string" }] }</pre>
401	Unauthorized
404	Not Found

Beschreibung der Datenfelder

Datenfeld	Definition	Bemerkung
CatalogueId	PK, Guid, not null	Eindeutige ID des Katalogs
CatalogueName	string, not null	Katalogname
VersionNumber	long, not null	Versionsnummer des Katalogs

Datenfeld	Definition	Bemerkung
VersionText	string, not null	Versionsnummer des Katalogs als Text
Description	string, null	Beschreibung zum Katalog
ModifiedBy	string, null	Benutzername desjenigen der die letzte Änderung durchgeführt hat
ModifiedDate	date, null	Datum der letzten Änderung
DimensionUnits	string, null	Die Einheiten die in diesem Katalog verwendet werden

Beispiel:

In diesem Beispiel werden alle PIPEFAB Kataloge abgerufen, die in Projekten verwendet werden dürfen.

Request:

<https://pipefabconnect/api/catalogue/list>

Response:

```
{
  "catalogues": [
    {
      "catalogueName": "PIPEFAB4_TEST_ME",
      "versionNumber": 1,
      "versionText": "1",
      "description": "Beispielkatalog",
      "modifiedBy": "system",
      "modifiedDate": "2021-07-30T11:40:03.433",
      "catalogueId": "b39504d6-b229-4a55-92af-dc7429087a91",
      "dimensionUnits": "mm:2; deg:1; kg:3; EUR:2,3; Pc:0; bar:2; s:0; kN:2;
kg/dm3:3; x:3; Nm:0; Strength:0"
    }
  ]
}
```

3.4.2 GET api/catalogue/header/?id=

Mit Hilfe dieser Webmethode können die Kopfdaten eines PIPEFAB Kataloges abgerufen werden.
Als *Id* wird die *CatalogueId* des Katalogs angegeben zu dem die Kopfdaten abgerufen werden sollen.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/header

Retrieve the catalogue header data for the specified catalogue ID.

🔒 ⬆

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type application/json

Code	Description
200	OK
	Example Value Model
	<pre>{ "catalogueName": "string", "versionNumber": 0, "versionText": "string", "readonly": true, "status": true, "standardLanguage": "string", "description": "string", "descriptionId": 0, "modifiedBy": "string", "modifiedDate": "2025-12-11T15:50:53.489Z", "source": "string", "visible": true, "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "companyInit": "string", "referenceName": "string", "noDelete": true, "dimensionUnits": "string", "edition": "string" }</pre>
401	Unauthorized
404	Not Found

Beschreibung der Datenfelder

Datenfeld	Definition	Bemerkung
CatalogueId	PK, Guid, not null	Eindeutige ID des Katalogs
CatalogueName	string, not null	Katalogname

Datenfeld	Definition	Bemerkung
VersionNummer	long, not null	Versionsnummer des Katalogs
VersionText	string, not null	Versionsnummer des Katalogs als Text
Description	string, null	Beschreibung zum Katalog
DescriptionId		ID der Beschreibung, die auf die Übersetzungstabelle des Rohrteilekatalogs für Nicht-Standardsprachen verweist
ModifiedBy	string, null	Benutzername desjenigen der die letzte Änderung durchgeführt hat
ModifiedDate	date, null	Datum der letzten Änderung
DimensionUnits	string, null	Die Einheiten die in diesem Katalog verwendet werden
CompanyInit		
ReferenceName		Name des Quellkatalogs, wenn dieser Katalog importiert oder aus PIPEFAB V29 konvertiert wurde
NoDelete		
Edition		Enthält die PIPEFAB Edition der diesen Katalog erzeugt hat
Visible		wenn true, ist der Katalog für Anwender sichtbar
Source		Name der Quelle, wenn der Katalog von einem externen System importiert wurde
Readonly		wenn true, ist der Katalog schreibgeschützt
Status		Status des Katalogs z.B.: 0=current, 1=alt, nicht für neue Isometrien geeignet
StandardLanguage		Standardsprache des Katalogs

Beispiel:

In diesem Beispiel werden die Kopfdaten zu dem PIPEFAB Katalog mit der CatalogueId = b39504d6-b229-4a55-92af-dc7429087a91 abgerufen.

Request:

[https://pipefabconnect/api/catalogue/header/?id= b39504d6-b229-4a55-92af-dc7429087a91](https://pipefabconnect/api/catalogue/header/?id=b39504d6-b229-4a55-92af-dc7429087a91)

Response:

```
{
  "catalogueName": "PIPEFAB4_TEST_ME",
  "versionNumber": 1,
  "versionText": "1",
  "readonly": false,
  "status": false,
  "standardLanguage": "de",
  "description": "Beispielkatalog",
  "descriptionId": 0,
  "modifiedBy": "system",
  "modifiedDate": "2021-07-30T11:40:03.433",
  "source": "",
  "visible": true,
  "catalogueId": "b39504d6-b229-4a55-92af-dc7429087a91",
  "companyInit": "PBS-SD",
  "referenceName": "PIPEFAB4_TEST_ME",
  "noDelete": false,
  "dimensionUnits": "mm:2; deg:1; kg:3; EUR:2,3; Pc:0; bar:2; s:0; kN:2;
kg/dm3:3; x:3; Nm:0; Strength:0",
  "edition": "ME"
}
```

3.4.3 GET api/catalogue/find/projects/?id=

Mit Hilfe dieser Webmethode kann eine Liste der PIPEFAB-Projekte die den angegebenen Katalog verwenden, abgerufen werden.

Als *Id* wird die *CatalogueId* des Katalogs angegeben zu der die Projektliste erstellt werden soll.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/find/projects

Find all projects that use the catalogue with the specified catalogue ID.

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type

application/json

Code	Description
200	OK
	Example Value Model <pre>{ "catalogueName": "string", "versionNumber": 0, "versionText": "string", "readonly": true, "status": true, "standardLanguage": "string", "description": "string", "descriptionId": 0, "modifiedBy": "string", "modifiedDate": "2026-01-12T13:58:42.631Z", "source": "string", "visible": true, "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "companyInit": "string", "referenceName": "string", "noDelete": true, "dimensionUnits": "string", "edition": "string" }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel wird eine Liste der Projekte abgerufen, die den Katalog mit der CataloguePartId = „b32504d6-b229-4b55-91af-dc7429087e91“ verwenden

Request:

<https://api.dev.pipfab.cloud/api/catalogue/find/projects/?id=b32504d6-b229-4b55-91af-dc7429087e91>

Response:

```
{
  "catalogueId": "b32504d6-b229-4b55-91af-dc7429087e91",
  "projects": [
    "78c004b1-3917-4dbb-9d0b-009569405911",
    "108a586f-c9b7-43fd-bfd7-1eb9c8160889",
    "5c504379-3d51-4ab3-b465-e0c12fce759b",
    "3568c313-951f-41fe-a500-e600c24f1ddd"
  ]
}
```

3.4.4 GET api/catalogue/connectortypes/?id=

Mit Hilfe dieser Webmethode können die definierten Verbindungstypen eines PIPEFAB Kataloges abgerufen werden.

Als *Id* wird die *CatalogueId* des Katalogs angegeben zu dem die Verbindungstypen abgerufen werden sollen.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/connectortypes
 Find all connector types for the catalogue with the specified catalogue ID.

🔒 ⬆

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type **application/json** ▾

Code	Description
200	OK
	Example Value Model <pre>{ "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "numberOfConnectorTypes": 0, "connectorTypes": [{ "connectorTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "connectorTypeName": "string", "autoplaceFlag": true, "description": "string", "descriptionId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "documentsCount": 0, "nominalDiameterUnit": 0, "od0Allowed": true, "spoolConnectFlag": true, "turnConnectedFlag": true, "partConnectorAssemblyType": 0 }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die Verbindungstypen zu dem PIPEFAB Katalog mit der CatalogueId = b39504d6-b229-4a55-92af-dc7429087a91 abgerufen.

Request:

<https://pipefabconnect/api/catalogue/connectortypes/?id= b39504d6-b229-4a55-92af-dc7429087a91>

Response:

```
{
  "catalogueId": "b32504d6-b229-4a55-92af-dc7429087a91",
  "numberOfConnectorTypes": 8,
  "connectorTypes": [
    {
      "connectorTypeId": "33147c03-2400-4f91-bdea-34782d26271f",
      "connectorTypeName": "MJ",
      "autoplaceFlag": true,
      "description": "Mechanical join",
      "descriptionId": "00000000-0000-0000-0000-000000000000",
      "documentsCount": 0,
      "nominalDiameterUnit": 1,
      "od0Allowed": false,
      "spoolConnectFlag": true,
      "turnConnectedFlag": false,
      "partConnectorAssemblyType": 0
    },
    {
      ...
    }
  ]
}
```

3.4.5 GET api/catalogue/connectorcompatibilities/?id=

Mit Hilfe dieser Webmethode können die definierten Verbindungskompatibilitäten eines PIPEFAB Kataloges abgerufen werden.

Als *Id* wird die *CatalogueId* des Katalogs angegeben zu dem die Verbindungskompatibilitäten abgerufen werden sollen.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/connectorcompatibilities
 Find all connector compatibilities for the catalogue with the specified catalogue ID.

🔒 ^

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type

application/json

Code	Description
200	OK
	Example Value Model <pre>{ "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "numberOfConnectorTypes": 0, "connectorTypes": [{ "connectorTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "connectorTypeName": "string", "autoplaceFlag": true, "description": "string", "descriptionId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "documentsCount": 0, "nominalDiameterUnit": 0, "od0Allowed": true, "spoolConnectFlag": true, "turnConnectedFlag": true, "partConnectorAssemblyType": 0 }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die Verbindungskompatibilitäten zu dem PIPEFAB Katalog mit der CatalogueId = b39504d6-b229-4a55-92af-dc7429087a91 abgerufen.

Request:

[https://pipefabconnect/api/catalogue/connectorcompatibilities/?id= b39504d6-b229-4a55-92af-dc7429087a91](https://pipefabconnect/api/catalogue/connectorcompatibilities/?id=b39504d6-b229-4a55-92af-dc7429087a91)

Response:

```
{
  "catalogueId": "b32504d6-b229-4a55-92af-dc7429087a91",
  "numberOfCompatibilities": 10,
  "connectorCompatibilityEntries": [
    {
      "connectorTypeId1": "b32504d6-b229-4b55-91af-dc7429087e91",
      "catalogueName1": "PIPEFAB4_TEST_ME",
      "catalogueId1": "b32504d6-b229-4b55-91af-dc7429087e91",
      "name1": "SCREW",
      "connectorTypeId2": "b32504d6-b229-4b55-91af-dc7429087e91",
      "catalogueName2": "PIPEFAB4_TEST_ME",
      "catalogueId2": "b32504d6-b229-4b55-91af-dc7429087e91",
      "name2": "MJ",
      "gasketFlag": false,
      "work": "",
      "createWeldSym": false,
      "weldConnectionType": 0
    },
    {
      ...
    }
  ]
}
```

3.4.6 GET api/catalogue/parttypes/?id=

Mit Hilfe dieser Webmethode können die definierten Teiletypen eines PIPEFAB Kataloges abgerufen werden.

Als *Id* wird die *CatalogueId* des Katalogs angegeben zu dem die Teiletypen abgerufen werden sollen.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET
/api/Catalogue/parttypes
Find all part types for the catalogue with the specified catalogue ID.

Parameters

Name	Description
id string(\$uuid) (query)	id

Responses
Response content type
application/json

Code	Description
200	OK Example Value Model <pre>{ "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "numberOfPartTypeHeaders": 0, "partTypeHeaderEntries": [{ "partType": "string", "category": "string", "imageName": "string", "partTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "exchangePartTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "partTypeRemark": "string", "showInplacementMenu": true, "defaultTemplateName": "string", "defaultSymbolId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "defaultTemplateId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "defaultModelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "defaultSymbolName": "string", "defaultModelName": "string", "partPartTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "partTypeInUse": true }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die Teiletypen zu dem PIPEFAB Katalog mit der CatalogueId = b39504d6-b229-4a55-92af-dc7429087a91 abgerufen.

Request:

<https://pipefabconnect/api/catalogue/parttypes/?id= b39504d6-b229-4a55-92af-dc7429087a91>

Response:

```
{
  "catalogueId": "b32504d6-b229-4a55-92af-dc7429087a91",
  "numberOfPartTypeHeaders": 13,
  "partTypeHeaderEntries": [
    {
      "partType": "ABZWEIGSTUTZEN",
      "category": "ABZWEIGSTUTZEN",
      "imageName": "InstallStutzen",
      "partTypeId": "dbf005c7-4044-4acc-a4b5-1929af6f97c2",
      "exchangePartTypeId": "00000000-0000-0000-0000-000000000000",
      "partTypeRemark": "",
      "showInPlacementMenu": true,
      "defaultTemplateName": "BOSS",
      "defaultSymbolId": "00000000-0000-0000-0000-000000000000",
      "defaultTemplateId": "15da2b0f-510f-5944-8c11-b457c5bc1fbb",
      "defaultModelId": "00000000-0000-0000-0000-000000000000",
      "defaultSymbolName": null,
      "defaultModelName": null,
      "partPartTypeId": "dbf005c7-4044-4acc-a4b5-1929af6f97c2",
      "partTypeInUse": true
    },
    {
      ...
    }
  ]
}
```

3.4.7 GET api/catalogue/partfatherandson/?id=

Mit Hilfe dieser Webmethode können die Vater-Sohn-Beziehungen der Teile eines PIPEFAB Kataloges abgerufen werden.

Als *Id* wird die *CatalogueId* des Katalogs angegeben zu dem die Vater-Sohn-Beziehungen abgerufen werden sollen.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/partfatherandson
Find all part father and son relations for the catalogue with the specified catalogue ID.

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type **application/json**

Code	Description
200	OK
	Example Value Model <pre>{ "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "numberOfPartFatherAndSons": 0, "fatherAndSonEntries": [{ "fatherArticle": "string", "fatherDescription": "string", "fatherCatalogueName": "string", "fatherPartId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "fatherCatId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "sonArticle": "string", "sonDescription": "string", "sonCatalogueName": "string", "sonPartId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "sonCatId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "substitutiontype": 0, "quantity": 0 }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die Vater-Sohn-Beziehungen der Teile zu dem PIPEFAB Katalog mit der CatalogueId = b39504d6-b229-4a55-92af-dc7429087a91 abgerufen.

Request:

<https://pipefabconnect/api/catalogue/partfatherandson/?id= b39504d6-b229-4a55-92af-dc7429087a91>

Response:

```
{
  "catalogueId": "b32504d6-b229-4a55-92af-dc7429087a91",
  "numberOfPartFatherAndSons": 10,
  "fatherAndSonEntries": [
    {
      "fatherArticle": "Bogen26,9x2.0-1.4571*",
      "fatherDescription": "",
      "fatherCatalogueName": "PIPEFAB4_TEST_ME",
      "fatherPartId": "0e60ab83-2ee3-4502-8ad4-fbd6ef2fb7c8",
      "fatherCatId": "b32504d6-b229-4b55-91af-dc7429087e91",
      "sonArticle": "Abzweigstutzen48.3x2.0-1.4571",
      "sonDescription": "",
      "sonCatalogueName": "PIPEFAB4_TEST_ME",
      "sonPartId": "3e63666d-2e6c-4000-a8dc-043e484103cd",
      "sonCatId": "b32504d6-b229-4b55-91af-dc7429087e91",
      "substitutiontype": 2,
      "quantity": 1
    },
    {
      ...
    }
  ]
}
```

3.4.8 GET api/catalogue/part/?id=

Mit Hilfe dieser Webmethode können die in PIPEFAB gespeicherten Katalogdaten zu einem Teil abgerufen werden. Die von PIPEFAB zurückgegebenen Daten bestehen zum einen aus allgemeinen Teiledaten (partData) und je nach Teiletype aus zusätzlichen Teiledaten (templateData)

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Catalogue/part GET api/catalogue/part/?id=....

🔒 ⤴

Parameters

Name	Description
id	id
string(\$uuid)	
(query)	

Responses

Response content type application/json

Code	Description
200	OK
	Example Value Model <pre>{ "partData": { "articleCode": "string", "mrpCode": "string", "versionNumber": 0, "versionText": "string", "status": 0, "description": "string", "modifiedBy": "string", "modifiedDate": "2025-12-02T09:03:30.594Z", "source": "string", "nominalPressure": 0, "noPlace": true, "materialName": "string", "standard": "string", "weight": 0, "templateName": "string", "shortDescription": "string", "groupType": "string", "price": 0, "dispo": 0, "placeholder": true, "cadFlag": true, "transferFlag": true, "comServSent": true, "isMaterial": true, "isLabour": true, "externalFatson": true, </pre>
401	Unauthorized

Beispiel:

In diesem Beispiel werden PIPEFAB Teiledaten für das Teil mit der CataloguePartId = „5044b381-c255-4303-be34-3640b99f311d“ gelesen. Bei dem Teil handelt es sich um einen Abzweigstutzen. Das Teil verfügt über zusätzliche Teiledaten aufgrund der Zuordnung zum Template „BOSS“.

Request:

<https://pipefabconnect/api/catalogue/part/?id=5044b381-c255-4303-be34-3640b99f311d>

Response:

```
{
  "partData": {
    "articleCode": "Abzweigstutzen12.0x1.0-1.4571",
    "mrpCode": "Abzweigstutzen88.9x3.0-1.4571",
    "versionNumber": 4,
    "versionText": "2 4",
    "status": 0,
    "description": null,
    "modifiedBy": null,
    "modifiedDate": null,
    "source": null,
    "nominalPressure": 0.00000,
    "noPlace": null,
    "materialName": "1.4571",
    "standard": null,
    "weight": 0.00000,
    "templateName": "BOSS",
    "shortDescription": null,
    "groupType": null,
    "price": 0.00000,
    "dispo": 0,
    "placeholder": null,
    "cadFlag": null,
    "transferFlag": null,
    "comServSent": null,
    "isMaterial": true,
    "isLabour": null,
    "externalFatson": null,
    "filename": null,
    "importDate": "2021-08-03T00:00:00",
    "createdBy": null,
    "createDate": "2021-08-03T00:00:00",
    "cadSymbol": null,
    "certification": null,
    "nominalDiameter1": 10.00000,
    "nominalDiameter2": 10.00000,
    "isStraight": true,
    "isCorner": false,
    "validByTemplate": true,
  }
}
```



```
"installGroupName": null,
"isTube": false,
"deleted": 0,
"partTreatment": 1,
"catPartId": "5044b381-c255-4303-be34-3640b99f311d",
"catalogueId": "b32504d6-b229-4b55-91af-dc7429087e91",
"partTypeId": "dbf005c7-4044-4acc-a4b5-1929af6f97c2",
"descriptionId": "00000000-0000-0000-0000-000000000000",
"symbolId": "bd07c009-691f-42c0-9cf6-988b32d03ab2",
"modelId": "00000000-0000-0000-0000-000000000000",
"allowAngleTrim": false,
"isExcentric": false,
"maxElbowAngle": 0.00000,
"isGasket": false,
"likeFlange": false,
"dimensionSequence": null,
"nominalDiameter": 10.00000,
"outerDiameter": 12.00000,
"wallThickness": 1.00000,
"wallThickness2": 1.00000,
"outerDiameter2": 12.00000,
"weightUnit": 0,
"pipefabBid": null,
"isForBranch": true,
"isTee": false,
"isLateralTee": false,
"nominalDiameter3": 0.00000,
"outerDiameter1": 12.00000,
"outerDiameter3": 0.00000,
"wallThickness1": 1.00000,
"wallThickness3": 0.00000,
"isAdditionalConnected": false,
"isAssignment": false,
"isPartProcessPart": false,
"supportsParametricDimensions": false,
"hasConnectorTypeDefined": true,
"hasConnectorGeometryDefined": true,
"mainBuildInLength": 37.00000
},
"templateData": {
  "connectorTypePc1Id": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
  "connectorTypePc2Id": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
  "dnAtPc1": 10.00000,
  "dnAtPc2": 10.00000,
  "heightA": 37.00000,
  "maxStraightTrimAtPc1": 0.00000,
  "odAtPc1": 12.00000,
  "odAtPc2": 12.00000,
  "thicknessAtPc1": 1.00000,
  "thicknessAtPc2": 1.00000,
  "bbhKey": null,
```



```
"catPartId": "5044b381-c255-4303-be34-3640b99f311d",  
"catalogueId": "b32504d6-b229-4b55-91af-dc7429087e91",  
"configurationCodeCfg": null,  
"componentId": null,  
"componentTypeCpt": null,  
"dimension0": 0.00000,  
"dimension1": 0.00000,  
"dimension2": 0.00000,  
"itemGroupCodeGrp": null,  
"symbolKey": null,  
"workPlan": null,  
"workPlan2": null,  
"workPlanComponent": null  
}  
}
```

3.5 Connector

3.5.1 GET api/connector/data/?id=

Mit Hilfe dieser Webmethode können zu einem Teil (CataloguePartId) die Anschlussdaten ermittelt werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Connector/data GET api/connector/data/?id=....

🔒 ^

Parameters

Name	Description
id	id
string(\$uuid)	
(query)	

Responses

Response content type application/json

Code	Description
200	OK
	Example Value Model
	<pre>{ "cataloguePartId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "connectorDataEntries": [{ "connectorName": "string", "connectorTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "cutMaximum": 0, "directionX": 0, "directionY": 0, "directionZ": 0, "dimensionBehaviour": "string", "extendPipe": 0, "isDimensionPoint": true, "nominalDiameter": 0, "nominalDiameterUnit": 0, "orientationUpX": 0, "orientationUpY": 0, "orientationUpZ": 0, "originX": 0, "originY": 0, "originZ": 0, "outerDiameter": 0, "wallThickness": 0 }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die PIPEFAB Anschlussdaten des Teils mit der CataloguePartId = „5044b381-c255-4303-be34-3640b99f311d“ gelesen.

Request:

<https://pipefabconnect/api/connector/data/?id=5044b381-c255-4303-be34-3640b99f311d>

Response:

```
{
  "cataloguePartId": "5044b381-c255-4303-be34-3640b99f311d",
  "connectorDataEntries": [
    {
      "connectorName": "PC1",
      "connectorTypeId": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
      "cutMaximum": 0.00000,
      "directionX": 0.00000,
      "directionY": -1.00000,
      "directionZ": 0.00000,
      "dimensionBehaviour": null,
      "extendPipe": 0.00000,
      "isDimensionPoint": false,
      "nominalDiameter": 10.00000,
      "nominalDiameterUnit": 0,
      "orientationUpX": -1.00000,
      "orientationUpY": 0.00000,
      "orientationUpZ": 0.00000,
      "originX": 0.00000,
      "originY": 0.00000,
      "originZ": 0.00000,
      "outerDiameter": 12.00000,
      "wallThickness": 1.00000
    },
    {
      "connectorName": "PC2",
      "connectorTypeId": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
      "cutMaximum": 0.00000,
      "directionX": 0.00000,
      "directionY": 1.00000,
      "directionZ": 0.00000,
      "dimensionBehaviour": null,
      "extendPipe": 0.00000,
      "isDimensionPoint": false,
      "nominalDiameter": 10.00000,
      "nominalDiameterUnit": 0,
      "orientationUpX": -1.00000,
      "orientationUpY": 0.00000,
      "orientationUpZ": 0.00000,
    }
  ]
}
```



```
"originX": 0.00000,  
"originY": 37.00000,  
"originZ": 0.00000,  
"outerDiameter": 12.00000,  
"wallThickness": 1.00000  
}  
1  
}
```

3.5.2 GET api/connector/type/?id=

Mit Hilfe dieser Webmethode können zu einem Anschlusstyp (ConnectorTypeId) die Daten des Anschlusstypen ermittelt werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Connector/type GET api/connector/type/?id=....

🔒 ^

Parameters

Name	Description
id string(\$uuid) (query)	id

Responses

Response content type application/json

Code	Description
200	OK Example Value Model <pre>{ "connectorTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "autoplaceFlag": true, "connectorTypeName": "string", "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "description": "string", "descriptionId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "nominalDiameterUnit": 0, "od0Allowed": true, "spoolConnectFlag": true, "turnConnectedFlag": true, "partConnectorAssemblyType": 0 }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die PIPEFAB Daten des Anschlusstypen mit der ConnectorTypeld = „5054b6f9-6092-4a5f-a003-37bdaa6f2d07“ gelesen.

Request:

<https://pipefabconnect/api/connector/type/?id=5054b6f9-6092-4a5f-a003-37bdaa6f2d07>

Response:

```
{
  "connectorTypeId": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
  "autoplaceFlag": false,
  "connectorTypeName": "WELD",
  "catalogueId": "b32504d6-b229-4b55-91af-dc7429087e91",
  "description": null,
  "descriptionId": "00000000-0000-0000-0000-000000000000",
  "nominalDiameterUnit": 0,
  "od0Allowed": false,
  "spoolConnectFlag": false,
  "turnConnectedFlag": false,
  "partConnectorAssemblyType": 3
}
```

3.5.3 GET api/connector/compatibility/?id=

Mit Hilfe dieser Webmethode können die zu einem Anschlusstypen (ConnectorTypeld) kompatiblen Anschlusstypen ermittelt werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Connector/compatibility GET api/connector/compatibility/?id=...

🔒 ^

Parameters

Name	Description
id string(\$uuid) (query)	id

Responses

Response content type application/json

Code	Description
200	OK Example Value Model <pre>{ "connectorTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "connectorCompatibilityEntries": [{ "connectorTypeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "gasketFlag": true, "work": "string", "createWeldSym": true, "weldConnectionType": 0 }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die in PIPEFAB hinterlegten kompatiblen Anschlusstypen zum Anschlusstyp mit der ConnectorTypeId = „5054b6f9-6092-4a5f-a003-37bdaa6f2d07“ gelesen.

Request:

<https://pipefabconnect/api/connector/compatibility/?id=5054b6f9-6092-4a5f-a003-37bdaa6f2d07>

Response:

```
{
  "connectorTypeId": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
  "connectorCompatibilityEntries": [
    {
      "connectorTypeId": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
      "gasketFlag": false,
      "work": "",
      "createWeldSym": false,
      "weldConnectionType": 1
    },
    {
      "connectorTypeId": "784a9aed-40f1-0044-9f65-5e62ab1d2736",
      "gasketFlag": false,
      "work": "",
      "createWeldSym": false,
      "weldConnectionType": 1
    },
    {
      "connectorTypeId": "125a6afa-cd39-4030-9f16-b039668d7450",
      "gasketFlag": false,
      "work": "",
      "createWeldSym": false,
      "weldConnectionType": 1
    }
  ]
}
```

3.6 Material (Werkstoffe)

3.6.1 GET api/material/byname

Suchen eines PIPEFAB Werkstoffes anhand des Werkstoffnamens.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Material/byname

Get material by name

🔒 ⬆

Parameters

Name	Description
name	
string	
(query)	

Responses

Response content type

application/json

Code	Description
200	OK
Example Value Model <pre>{ "materialId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "materialName": "string", "materialDescription": "string", "density": 0, "materialGroup": "string", "isForPpw": true, "oldMaterialDescription": "string", "textRemark": "string", "tensileStrength": 0, "yieldStrength": 0, "uniformElongation": 0 }</pre>	
401	Unauthorized
404	Not Found

Beschreibung der Datenfelder

Datenfeld	Definition	Bemerkung
MaterialId	PK, Guid, not null	ID des Werkstoffes
Material	PK, string, not null	Werkstoffnummer
MaterialDescription	string, null	Werkstoffbeschreibung

Datenfeld	Definition	Bemerkung
MaterialGroup	string, null	Werkstoffgruppe
IsForPpw	bool, not null	Wenn true, ist dieser Werkstoff für Kunststoffrohrschweißungen im Falle einer Stumpfschweißung relevant
OldMaterialDescription	string, null	Alte Werkstoffbeschreibung
TextRemark	string, null	Bemerkung
Density	decimal(18,5), not null	Dichte [kg/dm ³]
TensileStrength	decimal(18,5), not null	Zugfestigkeit [N/mm ²]
YieldStrength	decimal(18,5), not null	Streckgrenze [N/mm ²]
UniformElongation	decimal(18,5), not null	Gleichmaßdehnung [%]

Beispiel:

In diesem Beispiel wird der Werkstoffnummer „1.0308“ gesucht und gefunden.

Request:

GET <https://pipefabconnect/api/material/byname?name=1.0308>

Response (Auszug):

```
{
  "materialId": "064005f0-08a6-480c-80d5-03fb73193b80",
  "material": "1.0308",
  "materialDescription": "E235",
  "density": 7.85000,
  "materialGroup": "Stahl",
  "isForPpw": false,
  "oldMaterialDescription": "St35",
  "textRemark": "",
  "tensileStrength": 370.00000,
  "yieldStrength": 235.00000,
  "uniformElongation": 0.18000
}
```

3.6.2 GET api/material/list

Lesen der PIPEFAB Werkstoffdaten.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Material/list

Get the PIPEFAB material list

🔒 ⬆

Parameters

No parameters

Responses

Response content type

application/json

Code	Description
200	OK
	Example Value Model <pre>[{ "material": "string", "materialDescription": "string", "density": 0, "materialGroup": "string", "isForPpw": true, "oldMaterialDescription": "string", "textRemark": "string", "tensileStrength": 0, "yieldStrength": 0, "uniformElongation": 0 }]</pre>
401	Unauthorized
404	Not Found

Beschreibung der Datenfelder

Datenfeld	Definition	Bemerkung
MaterialId	PK, Guid, not null	ID des Werkstoffes
Material	PK, string, not null	Werkstoffnummer
MaterialDescription	string, null	Werkstoffbeschreibung
MaterialGroup	string, null	Werkstoffgruppe
IsForPpw	bool, not null	Wenn true, ist dieser Werkstoff für Kunststoffrohrschweißungen im Falle einer Stumpfschweißung relevant
OldMaterialDescription	string, null	Alte Werkstoffbeschreibung

Datenfeld	Definition	Bemerkung
TextRemark	string, null	Bemerkung
Density	decimal(18,5), not null	Dichte [kg/dm ³]
TensileStrength	decimal(18,5), not null	Zugfestigkeit [N/mm ²]
YieldStrength	decimal(18,5), not null	Streckgrenze [N/mm ²]
UniformElongation	decimal(18,5), not null	Gleichmaßdehnung [%]

Beispiel:

In diesem Beispiel werden alle vorhandenen PIPEFAB Werkstoffe gelesen.

Request:

GET <https://pipefabconnect/api/material/list>

Response (Auszug):

```
[
  {
    "materialId": "064005f0-08a6-480c-80d5-03fb73193b80",
    "material": "1.0308",
    "materialDescription": "E235",
    "density": 7.85000,
    "materialGroup": "Stahl",
    "isForPpw": false,
    "oldMaterialDescription": "St35",
    "textRemark": "",
    "tensileStrength": 370.00000,
    "yieldStrength": 235.00000,
    "uniformElongation": 0.18000
  },
  {
    ...
  }
]
```

3.6.3 POST api/material/add

Anlegen eines neuen Materials (Werkstoffes).

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus. Zu der Benutzer über die entsprechenden PIPEFAB-Rechte verfügen um einen Werkstoff anlegen zu können.

POST
/api/Material/add
 Add new material

Name	Description
body object (body)	Example Value Model <pre> { "materialName": "string", "materialDescription": "string", "density": 0, "materialGroup": "string", "isForPpw": true, "oldMaterialDescription": "string", "textRemark": "string", "tensileStrength": 0, "yieldStrength": 0, "uniformElongation": 0 } </pre> Parameter content type application/json

Responses

Response content type

application/json

Code	Description
200	OK Example Value Model <pre> [{ "materialId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "materialName": "string", "materialDescription": "string", "density": 0, "materialGroup": "string", "isForPpw": true, "oldMaterialDescription": "string", "textRemark": "string", "tensileStrength": 0, "yieldStrength": 0, "uniformElongation": 0 }] </pre>
400	Bad Request
401	Unauthorized

Beschreibung der Request-Datenfelder

Datenfeld	Definition	Bemerkung
Material	PK, string(80), not null	Werkstoffnummer
MaterialDescription	string(1024), null	Werkstoffbeschreibung
MaterialGroup	string(80), null	Werkstoffgruppe
IsForPpw	bool, not null	Wenn true, ist dieser Werkstoff für Kunststoffrohrschweißungen im Falle einer Stumpfschweißung relevant
OldMaterialDescription	string(1024), null	Alte Werkstoffbeschreibung
TextRemark	string(1024), null	Bemerkung
Density	decimal(18,5), not null	Dichte [kg/dm ³]
TensileStrength	decimal(18,5), not null	Zugfestigkeit [N/mm ²]
YieldStrength	decimal(18,5), not null	Streckgrenze [N/mm ²]
UniformElongation	decimal(18,5), not null	Gleichmaßdehnung [%]

Die Datenfelder mit der Definition „not null“ müssen im Request mit einem gültigen Wert gefüllt sein damit der Request akzeptiert wird. Die Anzahl der möglichen Zeichen im Datentyp String steht hinter dem Datentyp in den Klammern. Beispiel: string(80) -> es sind maximal 80 Zeichen zulässig.

Beschreibung der Response-Datenfelder

Datenfeld	Definition	Bemerkung
MaterialId	PK, Guid, not null	ID des Werkstoffes
Material	PK, string, not null	Werkstoffnummer
MaterialDescription	string, null	Werkstoffbeschreibung
MaterialGroup	string, null	Werkstoffgruppe
IsForPpw	bool, not null	Wenn true, ist dieser Werkstoff für Kunststoffrohrschweißungen im Falle einer Stumpfschweißung relevant
OldMaterialDescription	string, null	Alte Werkstoffbeschreibung
TextRemark	string, null	Bemerkung
Density	decimal(18,5), not null	Dichte [kg/dm ³]

Datenfeld	Definition	Bemerkung
TensileStrength	decimal(18,5), not null	Zugfestigkeit [N/mm ²]
YieldStrength	decimal(18,5), not null	Streckgrenze [N/mm ²]
UniformElongation	decimal(18,5), not null	Gleichmaßdehnung [%]

Beispiel:

In diesem Beispiel wird ein neuer Werkstoff mit dem Materialnamen (Werkstoffnummer) „TestMaterial“ angelegt.

Request:

POST <https://pipefabconnect/api/material/add>

```
{
  "materialName": "TestMaterial",
  "materialDescription": "TestMaterial angelegt über WebApi",
  "density": 1.0,
  "materialGroup": "Stahl",
  "isForPpw": false,
  "oldMaterialDescription": "St35",
  "tensileStrength": 1.0,
  "yieldStrength": 1.0,
  "uniformElongation": 1.0
}
```

Wenn das Material erfolgreich angelegt wurde, wird der angelegte Datensatz inkl. der vergebenen MaterialId zurückgeliefert.

Response:

```
{
  "materialId": "064004f2-08a6-480c-80d5-03fb73193b80",
  "material": "TestMaterial",
  "materialDescription": "TestMaterial angelegt über WebApi",
  "density": 1.00000,
  "materialGroup": "Stahl",
  "isForPpw": false,
  "oldMaterialDescription": "St35",
  "textRemark": "",
  "tensileStrength": 1.00000,
  "yieldStrength": 1.00000,
  "uniformElongation": 1.00000
}
```



Im Falle eines Fehlers, wie z.B. eines Validierungsfehlers wird eine Meldung zurückgeliefert.

```
Body Cookies Headers 4 Test Results  400 Bad Request • 271 ms • 194 B  
{ } JSON ▾ ▶ Preview  Debug with AI ▾  
1 "The material number is longer than 80 characters."
```

3.7 PieceOfPipe

3.7.1 POST api/pieceofpipe/job/result

Lesen der Piece of Pipe-Daten zu einem importierten Job.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

POST
/api/PieceOfPipe/job/result
 Post api/pieceofpipe/job/result

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
{
  "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "drawingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

Responses

Code	Description	Links
200	OK Media type application/json Controls Accept header. Example Value Schema <pre>{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "drawingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "productionUnits": [{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "name": "string", "materialGroup": "string", "materiallist": "string", "nominaldiameter": 0, "inchMeter": 0, "weight": 0, "plannedWorklocationName": "string", "pieceOfPipeList": [{ "productionPartId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "ideallength": 0, "cutlength": 0, "barcodeString": "string", "guideWayName": "string", "guideWayDescription": "string", "guideWaySequenceNumber": 0 }] }] }</pre>	No links
401	Unauthorized	No links
404	Not Found	No links

Beispiel:

In diesem Beispiel werden die PieceOfPipe-Daten zu einem erfolgreich importierten-Job abgefragt.

Job Id = "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae"

Request:

<https://pipefabconnect/api/pieceofpipe/job/result>

Request-Body:

```
{
  "jobId": "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae",
  "drawingId": "b6f0d019-aaaa-4946-9347-6f11d23765f0"
}
```

Response:

```
{
  "jobId": "9ebd106f-33d7-45e8-aaf7-ced73d0ce9ae",
  "drawingId": "b6f0d019-aaaa-4946-9347-6f11d23765f0",
  "productionUnits": [
    {
      "id": "41624d7c-23f1-437b-b52e-a1e11d90d803",
      "name": "01",
      "materialGroup": "",
      "materialList": "",
      "nominaldiameter": 40.00000,
      "inchMeter": 0.74922,
      "weight": 0.77151,
      "plannedWorkLocationName": "",
      "pieceOfPipeList": [
        {
          "productionPartId": "d8334a7d-7b17-4469-ad7f-39749952c265",
          "idealLength": 310.00000,
          "cutLength": 304.00000,
          "barcodeString": "020015661518073574501711261806399931204000",
          "guideWayName": "NurSägen",
          "guideWayDescription": "NurSägen",
          "guideWaySequenceNumber": 1
        }
      ]
    }
  ]
}
```

3.8 Production Unit

3.8.1 POST api/productionunit/job/result

Lesen der Fertigungseinheiten (production units) zu einem importierten Job.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

POST
/api/ProductionUnit/job/result
Get a list of production units by jobId and drawingId

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
{
  "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "drawingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

Responses

Code	Description	Links
200	OK Media type application/json Controls Accept header. Example Value Schema <pre>{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "drawingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "productionUnits": [{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "name": "string", "materialGroup": "string", "materialList": "string", "nominaldiameter": 0, "inchMeter": 0, "weight": 0, "plannedWorkLocationName": "string" }] }</pre>	No links
401	Unauthorized	No links
404	Not Found	No links

Beispiel:

In diesem Beispiel werden die Daten zu Fertigungseinheiten zu einem erfolgreich importierten-Job abgefragt. Job Id = "3f7e7aae-cfcc-4bf1-bb2e-6fa3a6dce7bb "

Request:

<https://pipefabconnect/api/productionunit/job/result>

Request-Body:

```
{
  "jobId": "3f7e7aae-cfcc-4bf1-bb2e-6fa3a6dce7bb",
  "drawingId": "6c00b12b-e4d5-4057-98fd-61cf9305ecb3"
}
```

Response:

```
{
  "jobId": "3f7e7aae-cfcc-4bf1-bb2e-6fa3a6dce7bb ",
  "drawingId": "6c00b12b-e4d5-4057-98fd-61cf9305ecb3",
  "productionUnits": [
    {
      "id": "41624d7c-23f1-437b-b52e-a1e11d90d803",
      "name": "01",
      "materialGroup": "",
      "materialList": "",
      "nominaldiameter": 40.00000,
      "inchMeter": 0.74922,
      "weight": 0.77151,
      "plannedWorkLocationName": "",
    }
  ]
}
```

3.8.2 POST api/productionunit/job/result/drawing

Abrufen des Vorschaubildes der Fertigungseinheiten (production units) zu einem importierten Job.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

POST

/api/ProductionUnit/job/result/drawing
 Get the drawing (png) of the given production unit id and job id




Parameters

Name	Description
body object (body)	Example Value Model <pre>{ "jobId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "productionUnitId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" }</pre> Parameter content type application/json

Responses

Response content type

application/json

Code	Description
200	OK Example Value Model <pre>"string"</pre>
400	Bad Request
401	Unauthorized

Beispiel:

In diesem Beispiel wird das Vorschaubild zu der angegebenen Fertigungseinheit zu dem zugehörigen, erfolgreich importierten-Job mit der Job Id = "047830c6-bc53-402a-85b5-db59d4eb5f06" abgefragt.

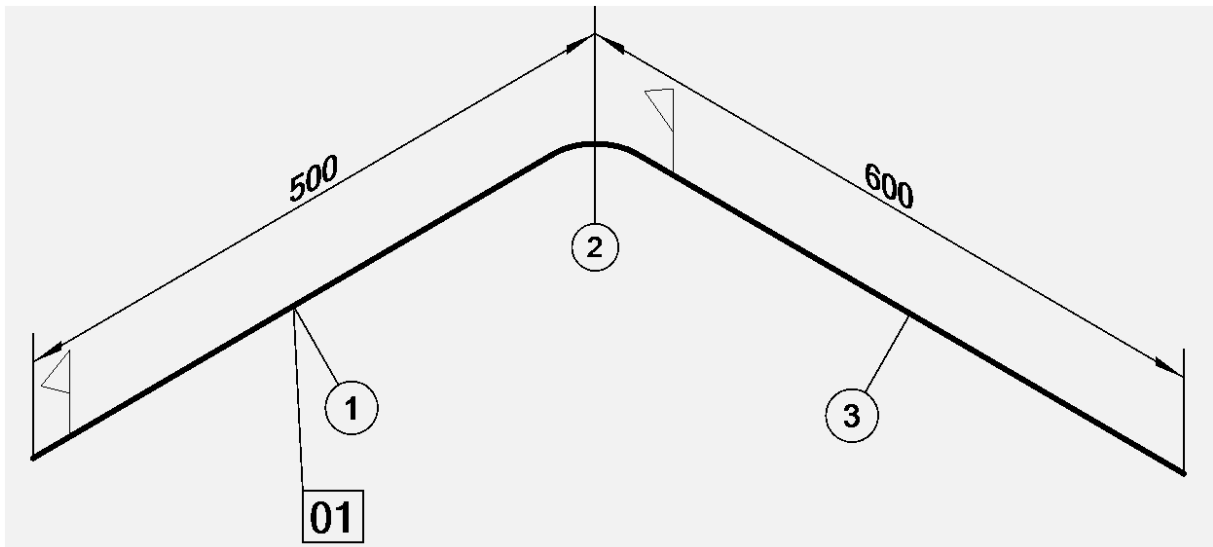
Request:

<https://pipefabconnect/api/productionunit/job/result/drawing>

Request-Body:

```
{
  "jobId": "047830c6-bc53-402a-85b5-db59d4eb5f06",
  "productionUnitId": "cefa86aa-c79c-4afd-8fb7-85a6ac8c5ecf"
}
```

Response:



3.9 Project

3.9.1 GET api/project/list

Lesen der vorhandenen Projekte. Geschlossene Projekte werden nicht berücksichtigt.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Project/list GET api/project/list

🔒 ⬆

Parameters

No parameters

Responses

Response content type application/json

Code	Description
200	OK Example Value Model <pre>{ "projects": [{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "name": "string" }] }</pre>
401	Unauthorized
404	Not Found



Beispiel:

Request:

<https://pipefabconnect/api/project/list>

Request-Body:

Response:

```
{
  "projects": [
    {
      "id": "108a586f-c9b7-43fd-bfd7-1eb9c8120389",
      "name": "PIPEFAB4_BATCH"
    },
    {
      "id": "5c504679-3d51-4ab3-b465-e0c121ce759b",
      "name": "PIPEFAB_CONNECT_TEST"
    },
    {
      "id": "3568c313-951f-41fe-a500-e600c24f1ddd",
      "name": "PIPEFAB4_TEST_ME"
    }
  ]
}
```

3.10 Project Components

3.10.1 GET api/projectcomponent/list

Lesen der Projektkomponenten zu einem Projekt.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET
/api/ProjectComponent/list
GET api/projectcomponent/list/?projectid=....

Parameters

Name	Description
projectid string(\$uuid) (query)	projectid

Responses
Response content type
application/json

Code	Description
200	OK
	Example Value Model <pre>{ "projectId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "numberOfComponents": 0, "components": [{ "catPartId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "catalogueName": "string", "catalogueVersion": "string", "catalogueDescription": "string", "articleCode": "string", "partType": "string", "templateName": "string", "specName": "string", "specDomain": "string", "specDescription": "string", "versionText": "string", "versionNumber": 0, "mrpCode": "string", "installGroupName": "string", "category": "string", "nominalPressure": 0, "nominalDiameter1": 0, "nominalDiameter2": 0, "nominalDiameter3": 0, "nominalDiameter4": 0, "materialName": "string", "standard": "string", }] }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die Projektkomponenten zur ProjectId „3568c313-951f-41fe-a500-e600c24f1ddd“ gelesen.

Request:

<https://pipefabconnect/api/projectcomponent/list/?projectid=3568c313-951f-41fe-a500-e600c24f1ddd>

Request-Body:Response:

Dargestellt ist nur ein Auszug (1. Komponente von 43)

```
{
  "projectId": "3568c313-951f-41fe-a500-e600c24f1ddd",
  "numberOfComponents": 43,
  "components": [
    {
      "catPartId": "5044b381-c255-4303-be34-3640b99f311d",
      "catalogueId": "b32504d6-b229-4b55-91af-dc7429087e91",
      "catalogueName": "PIPEFAB4_TEST_ME",
      "catalogueVersion": "1",
      "catalogueDescription": "Katalog für interne ME-Tests",
      "articleCode": "Abzweigstutzen12.0x1.0-1.4571",
      "partType": "ABZWEIGSTUTZEN",
      "templateName": "BOSS",
      "specName": "*",
      "specDomain": "*",
      "specDescription": "All Parts",
      "versionText": "2 4",
      "versionNumber": 4,
      "mrpCode": "Abzweigstutzen88.9x3.0-1.4571",
      "installGroupName": "[...]",
      "category": "ABZWEIGSTUTZEN",
      "nominalPressure": 0.00000,
      "nominalDiameter1": 10.00000,
      "nominalDiameter2": 10.00000,
      "nominalDiameter3": 0.00000,
      "nominalDiameter4": 0.00000,
      "materialName": "1.4571",
      "standard": "",
      "groupType": "",
      "certification": ""
    }
  ]
}
```

```

    "description": "",
    "isCorner": false,
    "isStraight": true,
    "isTube": false,
    "isExcentric": false,
    "isGasket": false,
    "isForBranch": true,
    "isTee": false,
    "isLateralTee": false,
    "isAdditionalConnected": false,
    "isAssignment": false,
    "noPlace": false,
    "outerDiameter1": 12.00000,
    "outerDiameter2": 12.00000,
    "outerDiameter3": 0.00000,
    "outerDiameter4": 0.00000,
    "wallThickness1": 1.00000,
    "wallThickness2": 1.00000,
    "wallThickness3": 0.00000,
    "wallThickness4": 0.00000,
    "mainBuildLength": 37.00000,
    "connectorTypeId1": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
    "connectorTypeId2": "5054b6f9-6092-4a5f-a003-37bdaa6f2d07",
    "connectorTypeId3": "00000000-0000-0000-0000-000000000000",
    "connectorTypeId4": "00000000-0000-0000-0000-000000000000",
    "pipeFabBid": "",
    "sketch": false
  },
  {
    (weitere Datensätze)
  }
]
}

```

3.11 State

3.11.1 GET api/state

Diese Webmethode teilt den aktuellen Status des PIPEFAB Connect Web Api's mit. Beim Aufruf dieser Webmethode werden folgende Prüfungen durchgeführt:

- Das PIPEFAB Connect Web Api prüft die Verbindung zur PIPEFAB Datenbank und ermittelt die PIPEFAB Datenbank Version. Diese wird als Ergebnis zurückgegeben.
- Das PIPEFAB Connect Web Api prüft die notwendige Lizenzierung und gibt das Resultat als Ergebnis zurück.
- Die Version des PIPEFAB Connect Web Api wird ermittelt und zurückgegeben.

GET
/api/State

Parameters

No parameters

Responses

Code	Description	Links
200	OK Media type application/json Controls Accept header. Example Value Schema <pre>{ "state": "string", "version": "string", "webApi": "string" }</pre>	No links
500	Internal Server Error	No links

Beispiel:

Anzeige des aktuellen Zustandes des PIPEFAB Connect Web Api's.

Request:

<https://pipefabconnect/api/state>

Response:

```
{
  "state": "OK",
  "version": "4.7.62.15001",
  "dbVersion": "805",
  "licenses": [
    "PIPEFAB Connect WebAPI PCF-Import",
    "PIPEFAB Connect WebAPI Kataloge",
    "PIPEFAB Connect WebAPI Projektkatalog"
  ],
  "webApi": "2"
}
```

Das Ergebnis zeigt das der Zustand des PIPEFAB Connect Web Api ok ist, eine gültige Lizenz existiert und es konnte eine Datenbankverbindung zur PIPEFAB Datenbank hergestellt werden. Die verwendete PIPEFAB Datenbankversion ist die 805. Die aktuelle Version des PIPEFAB Connect Web Api's ist die 4.7.62.15001. Die Angabe webApi = 2 wird für interne Zwecke verwendet und wird in Zukunft entfallen.

3.12 Symbol

3.12.1 GET api/symbol/data/?id=

Mit Hilfe dieser Webmethode können zu einem Teiletypen (PartTypeld) die Symboldaten ermittelt werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Symbol/data

Get the symbol data of the given symbol id

🔒 ^

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type

application/json

Code	Description
200	OK
	Example Value Model <pre>{ "symbolName": "string", "symbolGroup": "string", "symbolType": 0, "description": "string", "descriptionId": 0, "versionNumber": 0, "versionText": "string", "symbolId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "catalogueId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "templateId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "deleted": 0 }</pre>
401	Unauthorized
404	Not Found

Beispiel:

In diesem Beispiel werden die Symboldaten zu einem Teiletypen abgerufen. Der Teiletyp in diesem Beispiel hat die ID: bd07c009-691f-42c0-9cf6-988b32d03ab2 abgerufen.

Request:

<https://pipefabconnect/api/symbol/data/?id=bd07c009-691f-42c0-9cf6-988b32d03ab2>

Response:

```
{
  "symbolName": "BOSS",
  "symbolGroup": "Symbole",
  "symbolType": 0,
  "description": null,
  "descriptionId": 0,
  "versionNumber": 1,
  "versionText": "Geändert von system am 30.07.2021 13:42:01",
  "symbolId": "bd07c009-691f-42c0-9cf6-988b32d03ab2",
  "catalogueId": "b32504d6-b229-4b55-91af-dc7429087e91",
  "templateId": "15da2b0f-510f-5944-8c11-b457c5bc1fbb",
  "deleted": 0
}
```

3.12.2 GET api/symbol/drawing/?id=

Mit Hilfe dieser Webmethode kann zu einem Teiletypen (PartTypeld) das Symbolbild abgerufen werden.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

GET

/api/Symbol/drawing

Get the preview drawing (png) of the given symbol id

🔒 ^

Parameters

Name	Description
id	
string(\$uuid)	id
(query)	

Responses

Response content type application/json

Code	Description
200	OK
	Example Value Model
	"string"
400	Bad Request
401	Unauthorized

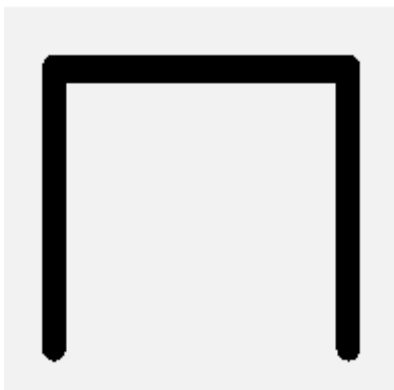
Beispiel:

In diesem Beispiel wird das Symbolbild zu einem Teiletypen abgerufen. Der Teiletyp in diesem Beispiel hat die ID: bd07c009-691f-42c0-9cf6-988b32d03ab2 abgerufen.

Request:

<https://pipefabconnect/api/symbol/drawing/?id=bd07c009-691f-42c0-9cf6-988b32d03ab2>

Response:



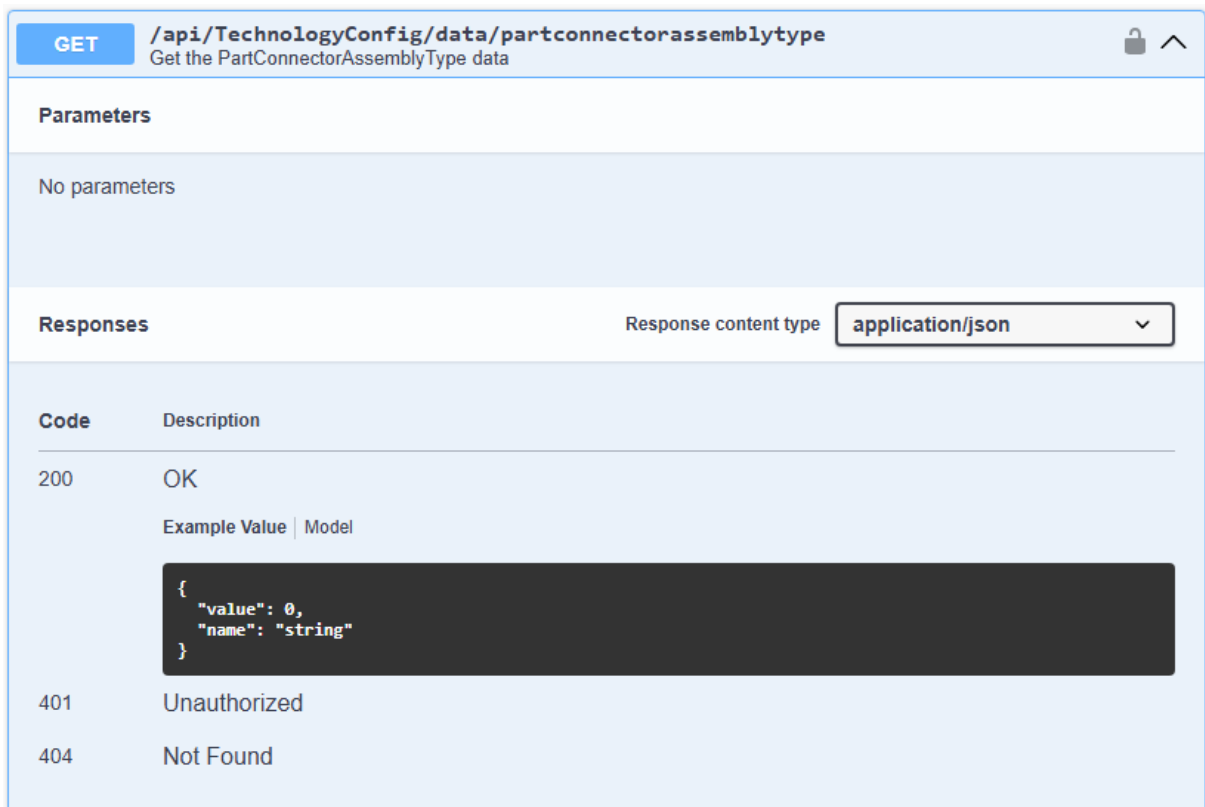
3.13 Technology Config Data (Lookup-Datentypen)

Die Webmethoden unterhalb der Url `api/technology/data/` liefern Daten der entsprechenden Lookup-Datentypen zurück. Dabei werden die jeweiligen Texte in der bevorzugten Sprache des jeweilig angemeldeten PIPEFAB-Benutzers zurückgegeben.

3.13.1 GET `api/technologyconfig/data/partconnectorassemblytype`

Lesen der Werte für den LoopUp-Typen *PartConnectorAssemblyType*. Verwendet wird dieser LookUp-Typ z.B. in den Daten des Anschlusstyps (`/api/connector/type/?id=?`).

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.



GET `/api/TechnologyConfig/data/partconnectorassemblytype`
Get the PartConnectorAssemblyType data

Parameters

No parameters

Responses Response content type: `application/json`

Code	Description
200	OK
	Example Value Model
	<pre>{ "value": 0, "name": "string" }</pre>
401	Unauthorized
404	Not Found

Beispiel:

Lesen der Werte für den LoopUp-Typen *PartConnectorAssemblyType*. Der PIPEFAB-Connect Benutzer hat als bevorzugte Sprache „de“ (Deutsch).

Request:

`https://pipefabconnect/api/technologyconfig/data/partconnectorassemblytype`

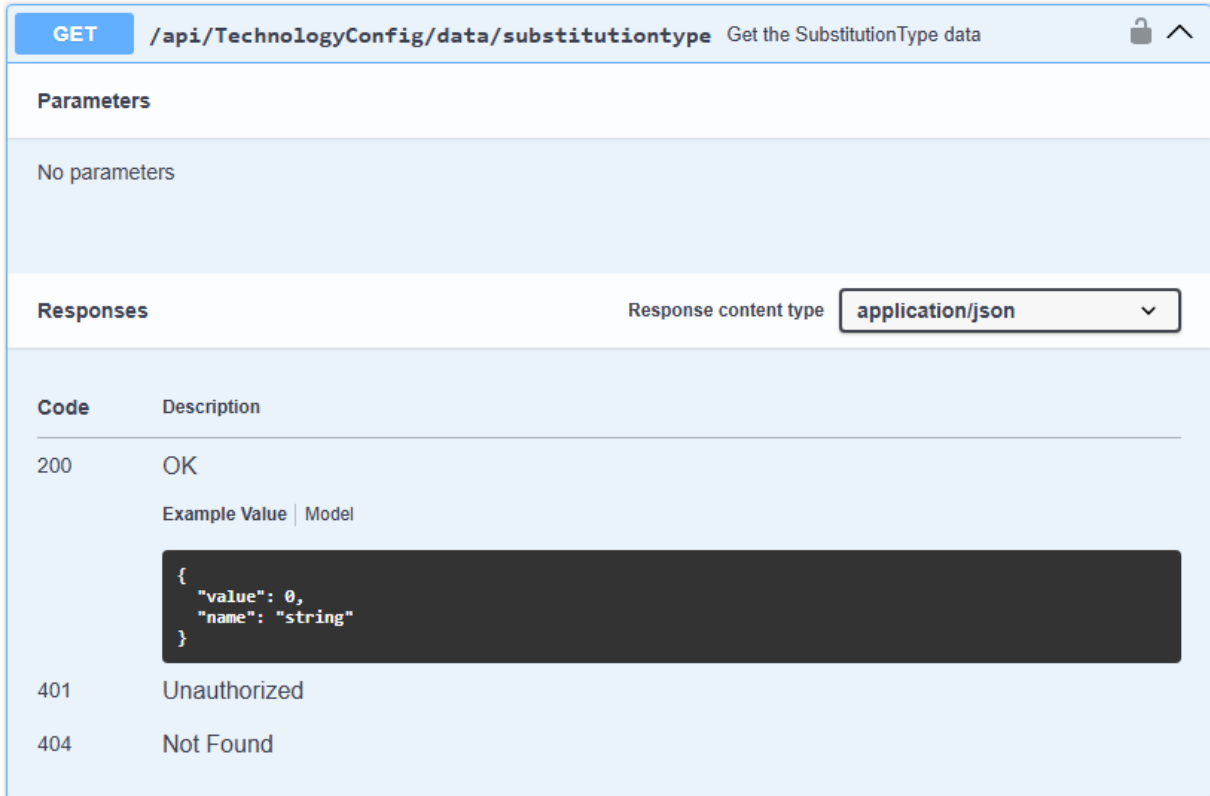
Response:

```
[
  {
    "value": 0,
    "name": "Andere"
  },
  {
    "value": 1,
    "name": "Geflanscht"
  },
  {
    "value": 2,
    "name": "Verschraubt, gesteckt oder andere lose Verbindung"
  },
  {
    "value": 3,
    "name": "Geschweißt, gelötet oder andere feste Verbindung"
  },
  {
    "value": 4,
    "name": "Offenes Rohrende"
  }
]
```

3.13.2 GET api/technologyconfig/data/substitutiontype

Lesen der Werte für den LoopUp-Typen *SubstitutionType*. Verwendet wird dieser LookUp-Typ z.B. in den Teile-Daten der Vater-Sohn-Beziehung (*/api/catalogue/partfatherandson/?id=?*).

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.



The image shows a Swagger UI interface for the endpoint `GET /api/TechnologyConfig/data/substitutiontype`. The interface includes a header with the method `GET` and the endpoint path. Below the header, there is a section for parameters, which is currently empty. The main section is for responses, with a dropdown menu for the response content type set to `application/json`. The response table lists three status codes: `200` (OK), `401` (Unauthorized), and `404` (Not Found). An example value for the `200` response is shown in a dark box, containing a JSON object: `{ "value": 0, "name": "string" }`.

Beispiel:

Lesen der Werte für den LoopUp-Typen *SubstitutionType*. Der PIPEFAB-Connect Benutzer hat als bevorzugte Sprache „de“ (Deutsch).

Request:

`https://pipefabconnect/api/technologyconfig/data/substitutiontype`

Response:

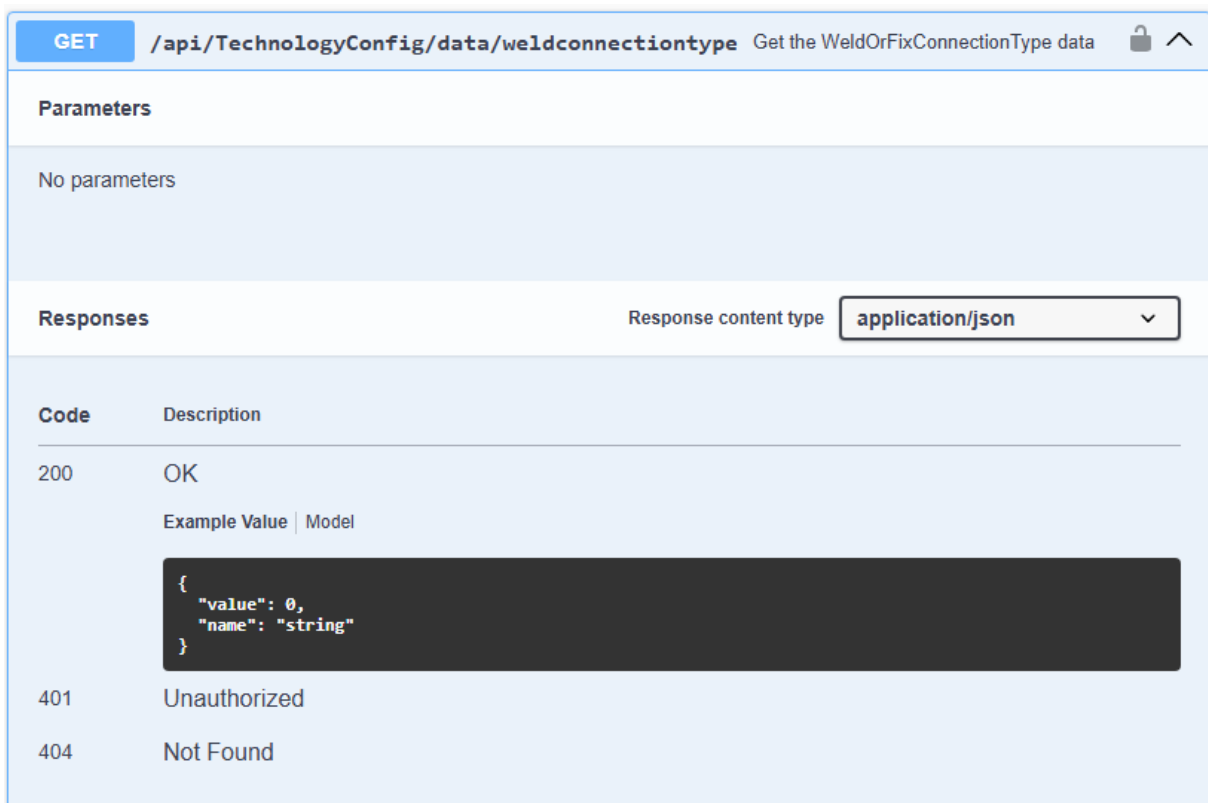
```
[
  {
    "value": 0,
    "name": "Immer"
  },
  {
```

```
[
  {
    "value": 1,
    "name": "Vorfertigung"
  },
  {
    "value": 2,
    "name": "Montage"
  }
]
```

3.13.3 GET api/technologyconfig/data/weldconnectiontype

Lesen der Werte für den LoopUp-Typen *WeldConnectionType*. Verwendet wird dieser LookUp-Typ z.B. in den Anschlusskompatibilitäten (*/api/catalogue/connectorcompatibilities/?id=?*).

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.



GET `/api/TechnologyConfig/data/weldconnectiontype` Get the WeldOrFixConnectionType data

Parameters

No parameters

Responses Response content type: `application/json`

Code	Description
200	OK
401	Unauthorized
404	Not Found

Example Value | Model

```
{
  "value": 0,
  "name": "string"
}
```

Beispiel:

Lesen der Werte für den LoopUp-Typen *WeldConnectionType*. Der PIPEFAB-Connect Benutzer hat als bevorzugte Sprache „de“ (Deutsch).

Request:

<https://pipefabconnect/api/technologyconfig/data/weldconnectiontype>

Response:

```
[
  {
    "value": 0,
    "name": "Andere Verbindung"
  },
  {
    "value": 1,
    "name": "Allgemein Schweißen"
  },
  {
    "value": 2,
    "name": "Autogenschweißen mit Acetylen"
  },
  {
    "value": 3,
    "name": "Manuelles WIG-Schweißen"
  },
  {
    "value": 4,
    "name": "Manuelles MIG-Schweißen"
  },
  {
    "value": 5,
    "name": "Manuelles MAG-Schweißen"
  },
  {
    "value": 6,
    "name": "Ummantelte Elektrode"
  },
  {
    "value": 7,
    "name": "Orbitalschweißmaschine"
  },
  {
    "value": 8,
    "name": "Allgemein Hartlöten"
  },
  {
    "value": 9,
    "name": "Allgemein Pressen"
  },
  {
    "value": 10,
    "name": "Allgemein Kleben"
  },
  {
    "value": 11,
```



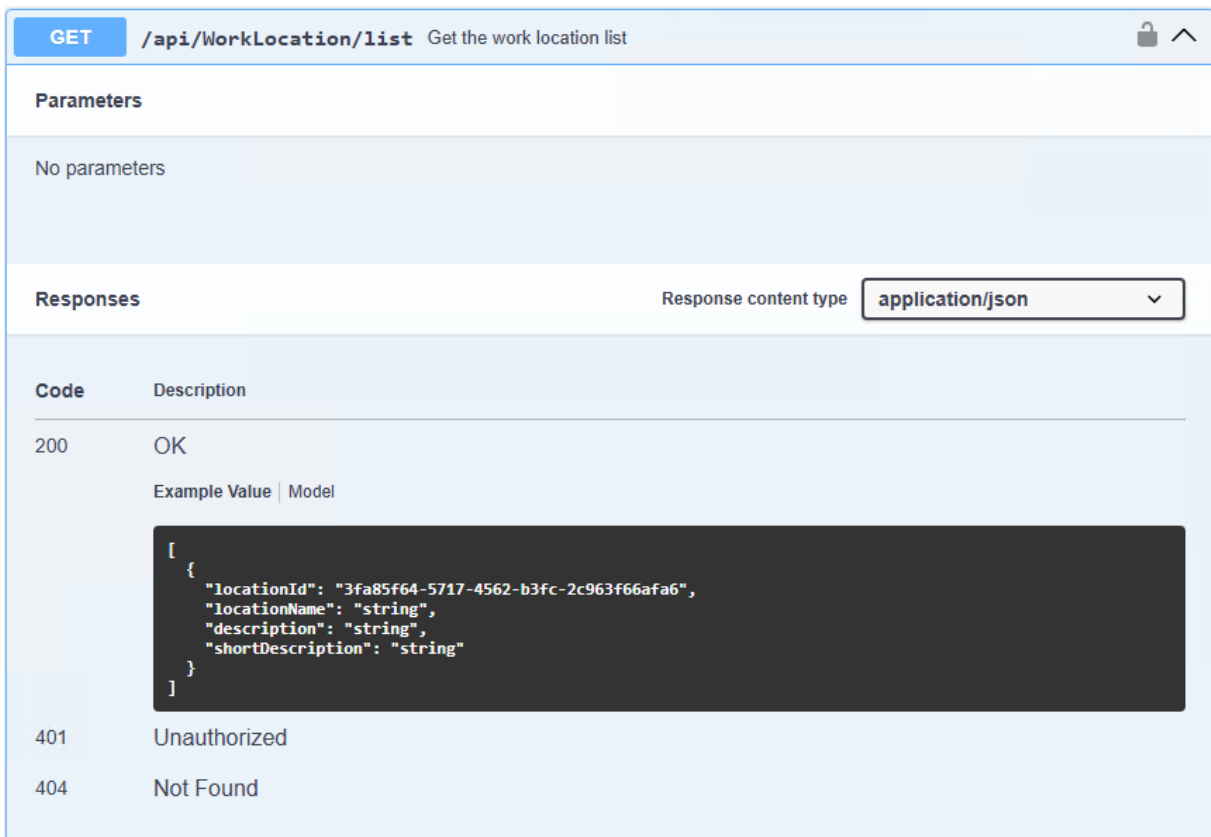
```
    "name": "Allgemein Weichlöten"  
  }  
]
```

3.14 Work Location (Fertigungsstätte)

3.14.1 GET api/worklocation/list

Lesen der Fertigungsstätten. Es wird eine Liste der vorhandenen Fertigungsstätten mit der zugehörigen LocationId bereitgestellt. Die LocationId der Fertigungsstätte wird für bei anderen Aufrufen des PIPEFAB Connect Web-API's benötigt. Informationen

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.



The image shows a Swagger UI interface for the endpoint `GET /api/WorkLocation/list`. The interface includes a header with the method `GET` and the endpoint path. Below the header, there is a section for parameters, which is currently empty. The main section is for responses, showing a table with status codes and descriptions. The first response is `200 OK`, with an example JSON response shown in a dark box. The other two responses are `401 Unauthorized` and `404 Not Found`.

Code	Description
200	OK
401	Unauthorized
404	Not Found

```
[
  {
    "locationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "locationName": "string",
    "description": "string",
    "shortDescription": "string"
  }
]
```

Beispiel:

Lesen der Fertigungsstätten.

Request:

`https://pipefabconnect/api/worklocation/list`

Response:



3.15 Work Package (Fertigungseinheit)

3.15.1 POST api/workpackage/isometricProcessingStatus

Anzeigen des aktuellen Bearbeitungsstands der Fertigungseinheiten einer Isometrie.

Der Aufruf setzt ein gültiges PIPEFAB Connect Sicherheitstoken voraus.

POST
/api/WorkPackage/isometricProcessingStatus
Get the processing status of an isometric drawing

Name	Description
body object (body)	Example Value Model <pre>{ "isometricName": "string", "locationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" }</pre> Parameter content type application/json

Responses

Response content type

application/json

Code	Description
200	OK Example Value Model <pre>[{ "isometricFullName": "string", "pipeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "version": 0, "workPackageName": "string", "workPackageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "puName": "string", "sequenceNo": "string", "orderQuantity": 0, "workflowStatus": 0, "annotationReady": true, "bendReady": true, "branchReady": true, "cleanReady": true, "cutReady": true, "endPreparationReady": true, "finalCutReady": true, "measureReady": true, "tackReady": true, "weldReady": true, "hasAnnotation": true, "hasBend": true, "hasBranch": true, "hasClean": true, "hasCut": true, "hasEndPreparation": true, "hasFinalCut": true, }]</pre>
401	Unauthorized
404	Not Found

Beschreibung der Request-Datenfelder

Datenfeld	Definition	Bemerkung
IsometricName	string, not null	Name der Isometrie zu der der Fertigungszustand abgefragt werden soll
LocationId	Guid , not null	Id der Fertigungsstätte

Die Datenfelder mit der Definition „not null“ müssen im Request mit einem gültigen Wert gefüllt sein damit der Request akzeptiert wird.

Gesucht wird der Isometrienamen zur angegebenen LocationId (Fertigungsstätte). Der Isometrienamen wird dabei in dem vollen Pfad der Isometrie gesucht. Dieser Pfad umfasst neben dem Isometrienamen auch das Projekt und die ggf. vorhandenen Ebenenstruktur.

Beispiel:

TestProjekt\Ebene1\Ebene2\Isometrienamen

Sollte keine Isometrie gefunden werden, wird ein Http-Fehler 404 (not found) zurückgegeben.

Beschreibung der Response-Datenfelder

Datenfeld	Definition	Bemerkung
IsometricFullName	string	Name der Isometrie inkl. Projekt und Ebene
PipeId	Guid	Die Id des Rohres
Version	int	Isometrieversion
WorkPackageName	string	Name der Fertigungsauftrag
WorkPackageId	Guid	Id des Fertigungsauftrags
PuName	string	Name der Fertigungseinheit
SequenceNo	string	lfd. Nummer des Spools zur Fertigungseinheit
OrderQuantity	long	Bestellmenge
WorkflowStatus	int	Aktueller Bearbeitungszustand
AnnotationReady	bool	Beschriftung fertig
BendReady	bool	Biegen fertig

Datenfeld	Definition	Bemerkung
BranchReady	bool	Aushalsen fertig
CleanReady	bool	Reinigen fertig
CutReady	bool	Zuschnitt fertig
EndPreparationReady	bool	Endbearbeitung fertig
FinalCutReady	bool	Endzuschnitt fertig
MeasureReady	bool	Biegemessung fertig
TackReady	bool	Heften fertig
WeldReady	bool	Schweißen fertig
HasAnnotation	bool	Prozessschritt Beschriften
HasBend	bool	Prozessschritt Biegen
HasBranch	bool	Prozessschritt Aushalsung
HasClean	bool	Prozessschritt Reinigen
HasCut	bool	Prozessschritt Zuschnitt
HasEndPreparation	bool	Prozessschritt Endbearbeitung
HasFinalCut	bool	Prozessschritt Endzuschnitt
HasMeasure	bool	Prozessschritt Biegemessung
HasTack	bool	Prozessschritt Heften
HasWeld	bool	Prozessschritt Schweißen

Bedeutung der Werte für den WorkflowStatus

Wert	Code	Bedeutung
0	Deleted	Die Fertigungseinheit wurde gelöscht
10	PipeLineCreated	Rohrleitung in R&I erzeugt
20	IsoCreated	Isometrie erzeugt
30	DesignReady	Konstruktion fertig
40	InPlanning	Materialbedarf geplant
50	ReleasedForFabrication	Für Anfertigung freigegeben
55	PlannedForOnSiteMeasurement	Geplant für Vor-Ort-Messung
60	PlannedForFabrication	Für Anfertigung geplant
70	FabricationStarted	Anfertigung begonnen
80	FabricationFinished	Anfertigung fertig
90	InDelivery	In Auslieferung
100	Delivered	Geliefert

Wert	Code	Bedeutung
110	AssemblyPlanned	Baustellenmontage geplant
120	AssemblyStarted	Baustellenmontage begonnen
130	AssemblyFinished	Baustellenmontage fertig
140	Invoiced	Fakturiert
150	Payed	Bezahlt

Beispiel:

Lesen der Fertigungsstätten.

Request:

<https://pipefabconnect/api/workpackage/isometricProcessingStatus>

```
{
  "isometricName": "CAD-PCF-ImportTest-1",
  "locationId": "00000000-0000-0000-0000-000000000001"
}
```

Response:

4 PIPEFAB Connect Client (NuGet-Package)

Der PIPEFAB Connect Client ermöglicht eine einfache Anbindung und Zugriff auf das PIPEFAB Connect WebApi. Der PIPEFAB Connect Client stellt dazu alle notwendigen Funktionen zur Verfügung. Dabei übernimmt der PIPEFAB Connect Client eigenständig das notwendige JWT-Handling. Der Anwender des PIPEFAB Connect Client muss lediglich die Login-Daten und die Url zum PIPEFAB Connect WebApi angeben.

Der PIPEFAB Connect Client führt im Hintergrund das Login durch und verwaltet das vom PIPEFAB Connect WebApi generierte Token.

Der PIPEFAB Connect Client wird als NuGet-Package zu Verfügung gestellt. Der PIPEFAB Connect Client ist kompatibel zu Microsoft .NET Standard 2 und Microsoft .NET6 ff.

4.1 Beispiel GetState

Das folgende Beispiel einer .NET Consolen-Anwendung ermittelt den aktuellen Status des PIPEFAB Connect WebApi's. Dazu wird die Web-Methode *GetState* des PIPEFAB Connect WebApi's aufgerufen.

Zum Ausführen des Beispiels muss lediglich die Url zum PIPEFAB Connect WebApi in der Variable *baseUrl* eingetragen werden und die gültigen Angaben für den *Username* und das *Passwort* ergänzt werden.

```
using System;
using System.Net;
using System.Threading.Tasks;

namespace PipeFab.PipeFabConnect.Client.Example.GetState
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            Console.WriteLine("PBS PIPEFAB Connect WebApi Example GetState\n");

            const string baseUrl = "https://????/";

            await GetState(baseUrl);

            Console.WriteLine("\nPlease press ENTER");
            Console.ReadLine();
        }

        private static async Task GetState(string baseUrl)
        {
            var client = new PipeFabConnectClient(new Contracts.Options.PipeFabConnectOption()
            {
                UseHandler = true,
                Username = "????",
                Password = "????",
                BaseUrl = baseUrl
            }, null, true);

            var response = await client.GetState();

            Console.WriteLine($"Response.StatusCode: {response.HttpStatusCode}\n");

            if (response.HttpStatusCode == HttpStatusCode.OK)
            {
                Console.WriteLine($"PIPEFAB Connect WebApi Version: {response.Result.Version}");
                Console.WriteLine($"PIPEFAB Connect WebApi State: {response.Result.State}");
                Console.WriteLine($"PIPEFAB Connect WebApi {response.Result.WebApi}");
            }
        }
    }
}
```

4.2 Beispiel Import PCF

Das folgende Beispiel einer .NET Consolen-Anwendung importiert eine PCF-Datei mit Hilfe des PIPEFAB Connect WebApi's. Dazu wird die Web-Methode *ImportPcf* des PIPEFAB Connect WebApi's aufgerufen.

Zum Ausführen des Beispiels muss lediglich die Url zum PIPEFAB Connect WebApi in der Variable *baseUrl* eingetragen werden und der *FileName* für die zu importierende PCF-Datei, der Namen der zu verwendenden *CadConSettings*, der *Username* und das *Passwort* ergänzt werden.

```
using PipeFab.PipeFabConnect.Contracts.Request;
using System;
using System.Configuration;
using System.IO;
using System.Net;
using System.Threading.Tasks;

namespace PipeFab.PipeFabConnect.Client.Example.ImportPcf
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            Console.WriteLine("PBS PIPEFAB Connect WebApi Example ImportPcf\n");

            const string baseUrl = "https://????/";

            await ImportPcf(baseUrl);

            Console.WriteLine("\nPlease press ENTER");
            Console.ReadLine();
        }

        private static async Task ImportPcf(string baseUrl)
        {
            var client = new PipeFabConnectClient(new Contracts.Options.PipeFabConnectOption()
            {
                UseHandler = true,
                Username = "????",
                Password = "????",
                BaseUrl = baseUrl
            }, null, true);

            var fileName = "????";

            var file = File.ReadAllBytes(fileName);

            var pcfJobRequest = new ImportJobRequest
            {
                FileBytes = file,
                FileName = fileName,
                CadConSettingName = "????",
                ImportMode = ImportMode.CheckAndSave
            };

            var response = await client.ImportPcfAsync(pcfJobRequest);

            Console.WriteLine($"Response.StatusCode: {response.HttpStatusCode}\n");

            if (response.HttpStatusCode == HttpStatusCode.OK)
            {
                Console.WriteLine($"Pcf import job successfully queued. JobId: {response.Result.JobId}");
            }
        }
    }
}
```

```
}
}
}
```

5 Installation

5.1 PIPEFAB Connect Web-API

Für die Installation des PIPEFAB-Connect Web-API's wird ein Server mit dem Microsoft Betriebssystem Windows Server 2022 oder neuer benötigt. Andere Windows Server Betriebssystemversionen sind nach Absprache möglich.

5.1.1 Systemvoraussetzungen

Diese Installationsanleitung basiert auf der Annahme, dass das PIPEFAB Connect Web-API auf einem Windows Server 2022 Standard (Desktop Experience) mit voreingestellter englischer Sprache installiert wird. Die folgende Abbildung zeigt die Versionsinformation des hier verwendeten Windows Server 2022.

About

Windows specifications

Edition	Windows Server 2022 Standard
Version	21H2
Installed on	16.11.2023
OS build	20348.2113

Abbildung 1: Versionsinformation

Bevor mit den vorbereitenden Maßnahmen zur Installation des PIPEFAB Connect Web-API's begonnen wird, sollten alle notwendigen Microsoft Sicherheitsupdates auf dem Server installiert sein. Diese gilt auch für zukünftige Microsoft Sicherheitsupdates.

5.1.2 Lizenzierung

Zum Betreiben des PIPEFAB Connect Web-API wird das Lizenzpaket *PIPEFAB 4 Connect SMAP 3D* benötigt. In diesem Lizenzpaket ist die benötigte Lizenz *PIPEFAB Connect WebAPI PCF-Import* enthalten.

5.1.3 Web Server IIS installieren

Für das PIPEFAB Connect Web-API wird der Internet Information Services benötigt. Dieser ist mit den Features gemäß der folgenden Abbildung zu installieren. Dazu das *Windows Server ControlPanel* starten und unter *Programs->Turn Windows feature on or off* auswählen. Die folgenden Screenshots zeigen den Ablauf der Installation der Internet Information Services.

Den Punkt *Role-based or feature-based Installation* auswählen (voreingestellt) und mit *Next* fortfahren.

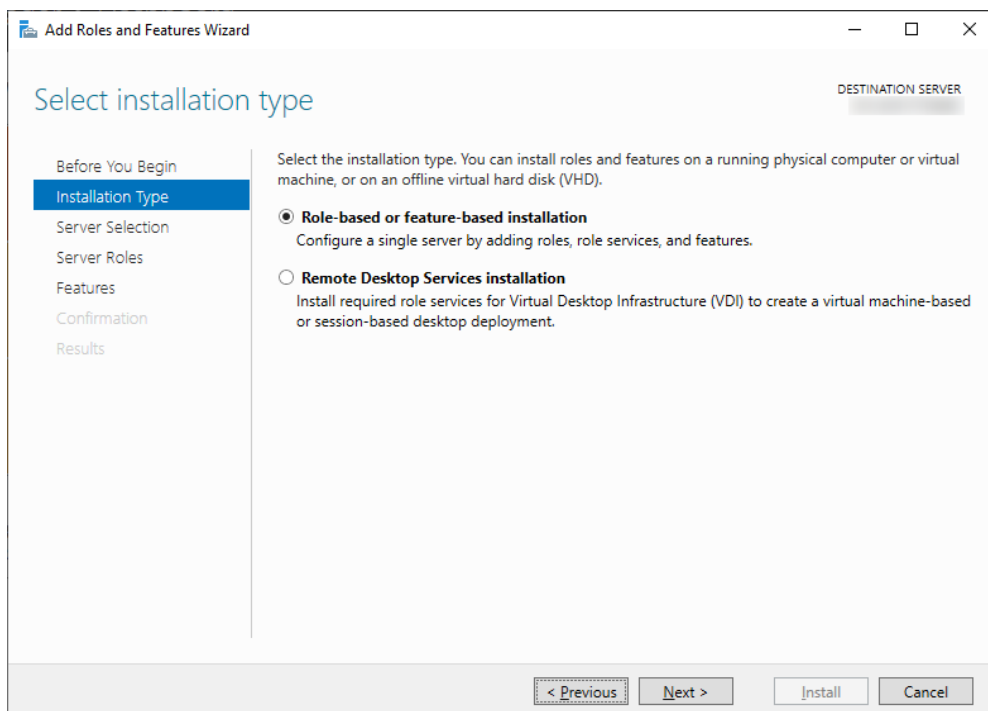


Abbildung 2: Start der Installation

Den aktuellen Server (voreingestellt) auswählen und mit *Next* fortfahren.

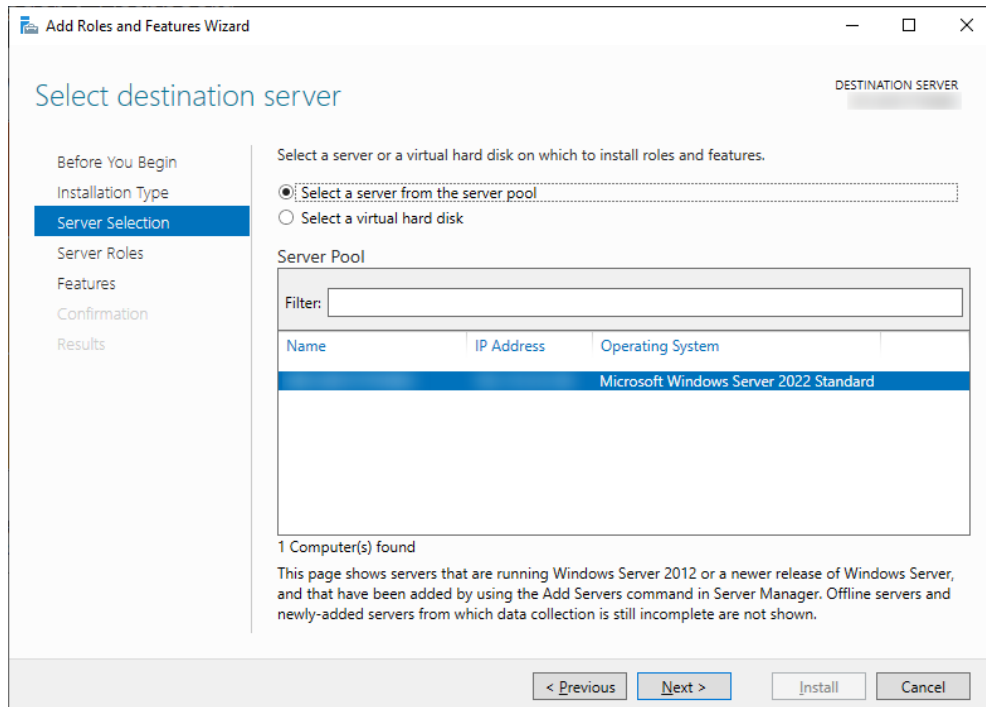


Abbildung 3: Server auswählen

In diesem Dialog bitte die Rolle *Web Server (IIS)* auswählen.

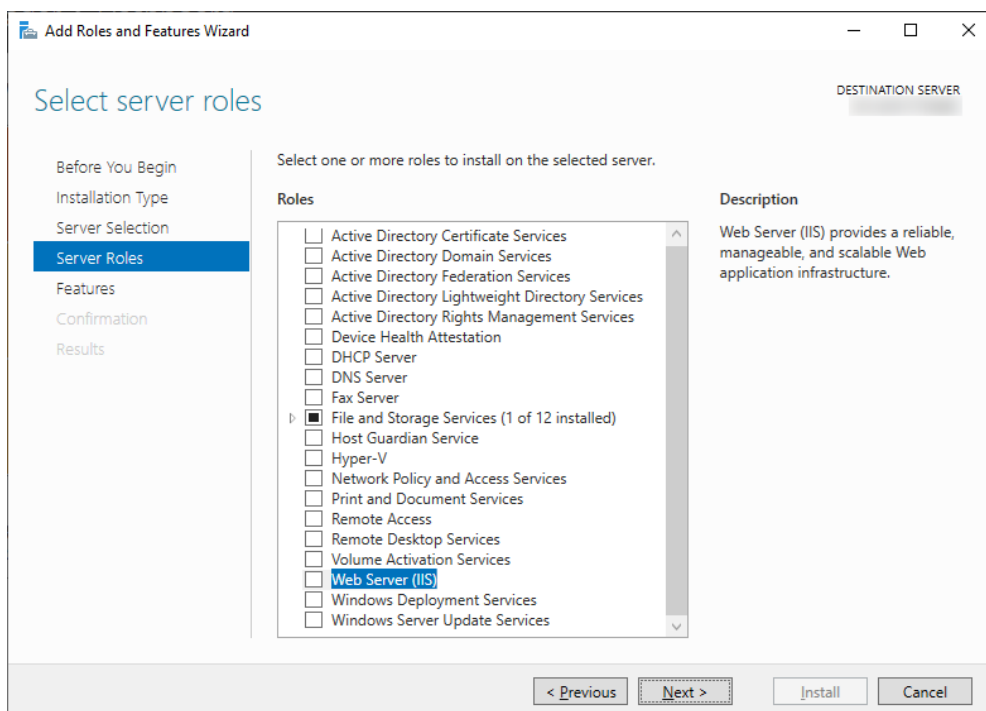


Abbildung 4: Roles

Beim Auswählen der Rolle *Web Server (IIS)* erscheint der folgende Dialog, der einfach durch den Button *Add Features* bestätigt wird.

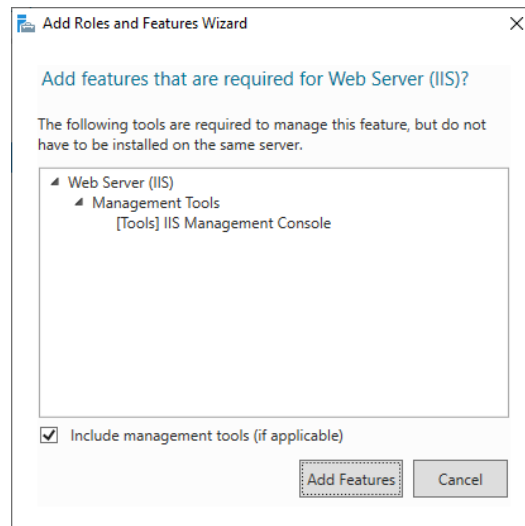


Abbildung 5: Feature Web Server IIS

Mit *Next* fortfahren.

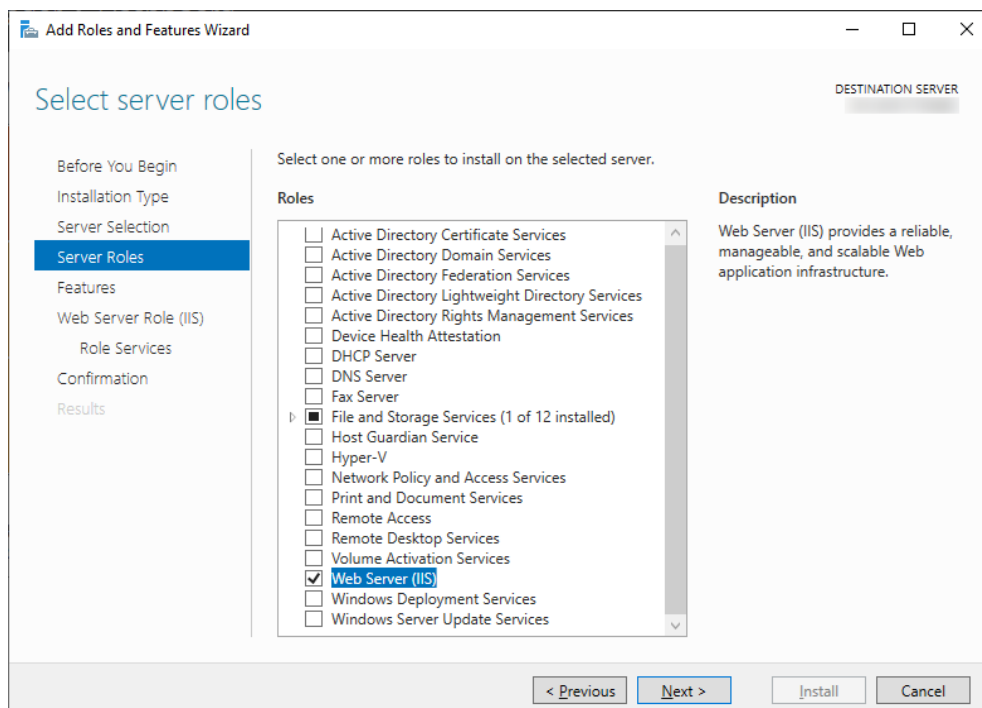


Abbildung 6: Server Role Web Server IIS

Mit *Next* fortfahren.

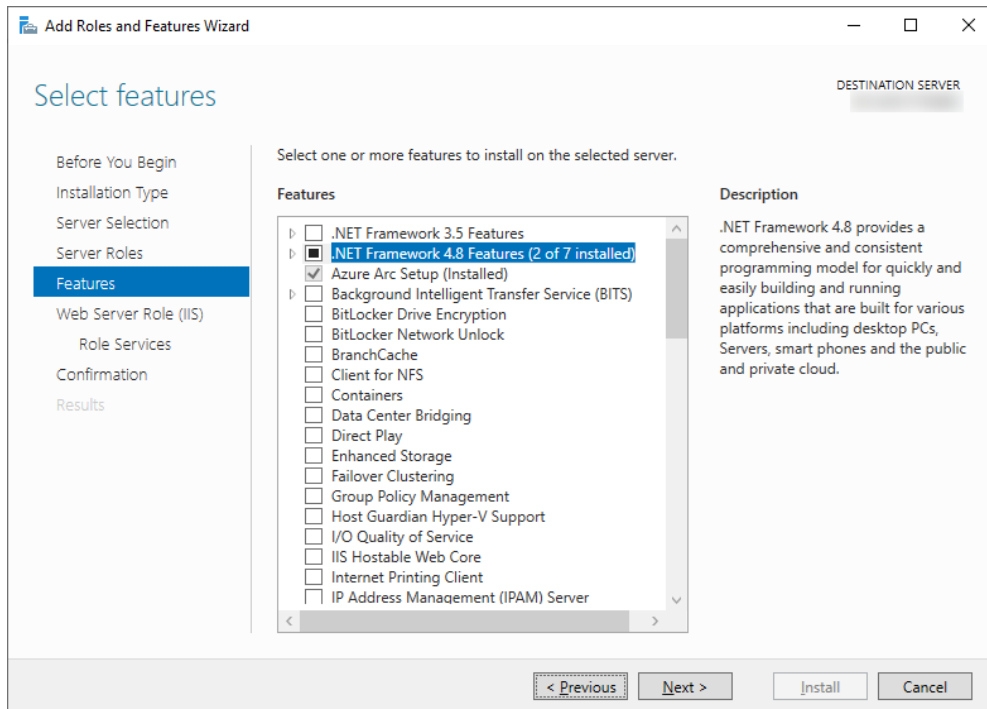


Abbildung 7: Installierte Features

Mit Next fortfahren.

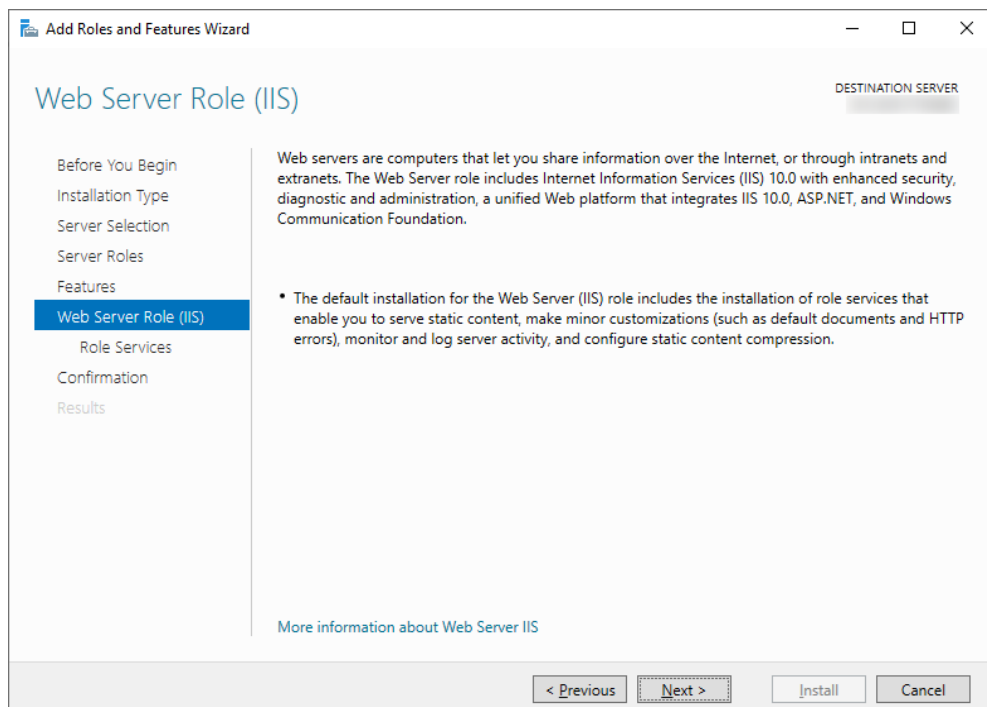


Abbildung 8: Hinweis zum Web Server (IIS)

Mit Next fortfahren.

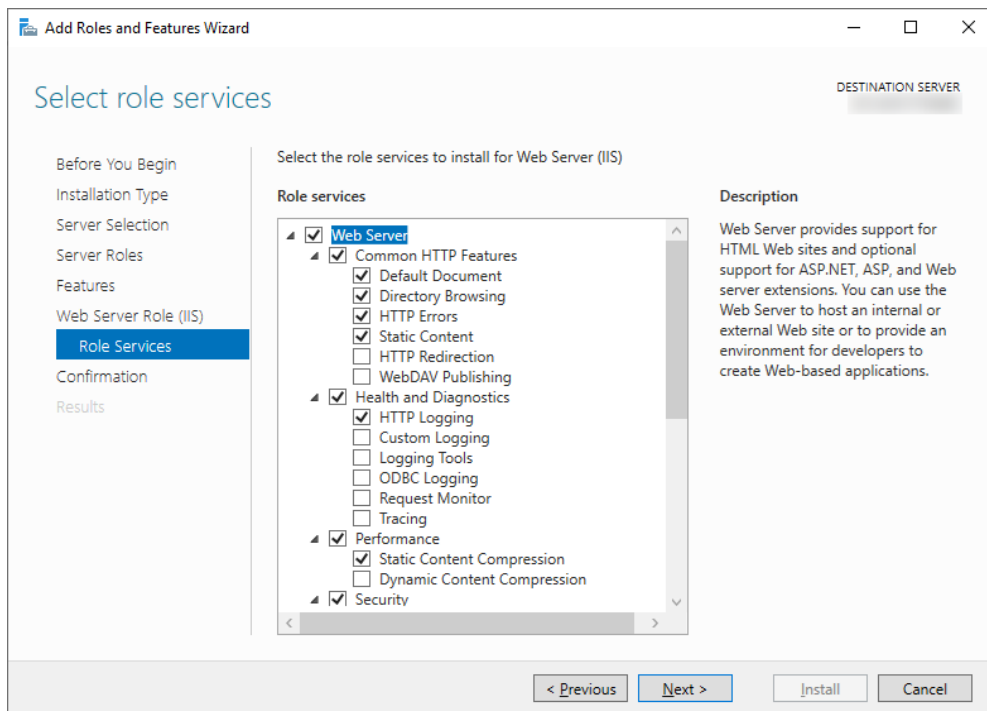


Abbildung 9: Role Services

Mit *Install* die Installation des Web Servers starten.

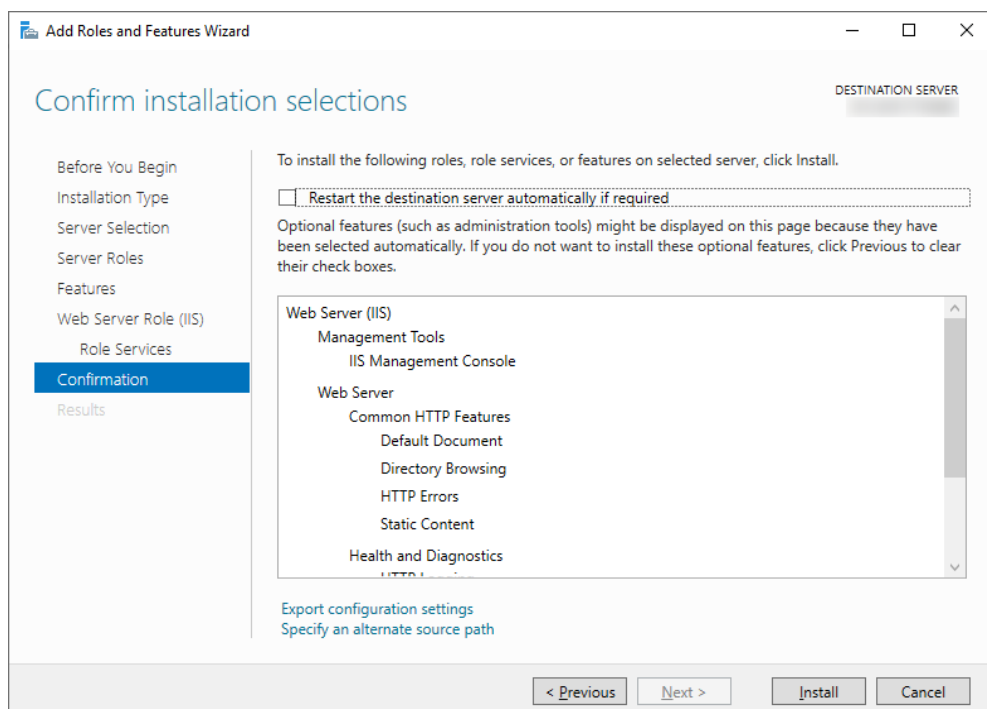


Abbildung 10: Bestätigung

Nach dem Abschluss den folgenden Dialog mit *Close* beenden. Damit ist der Web Server IIS installiert.

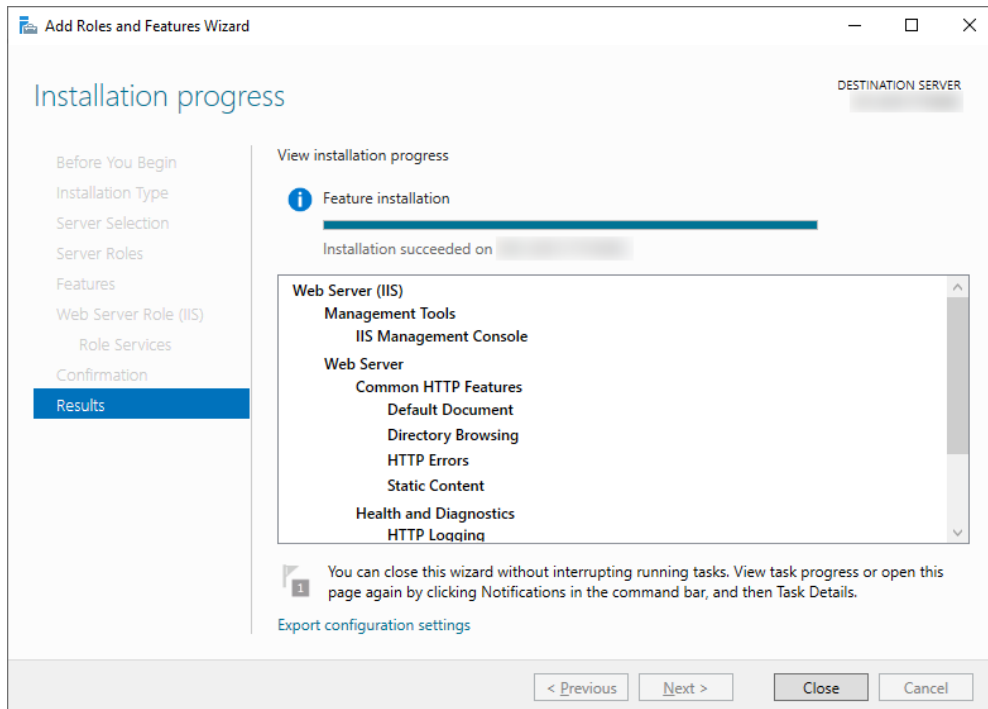


Abbildung 11: Installationsergebnis

5.1.4 ASP.NET Core Hosting Bundle für .NET 9 installieren

Im nächsten Schritt muss das Asp.Net Core Windows Hosting Bundle für .NET 9 installiert werden. Dazu bitte den Installer unter folgenden Link verwenden.

<https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-9.0.13-windows-hosting-bundle-installer>

und ausführen. Es öffnet sich der folgende Installationsdialog, der nach Annahme der Lizenzbedingungen mit *Install* fortgesetzt werden kann.

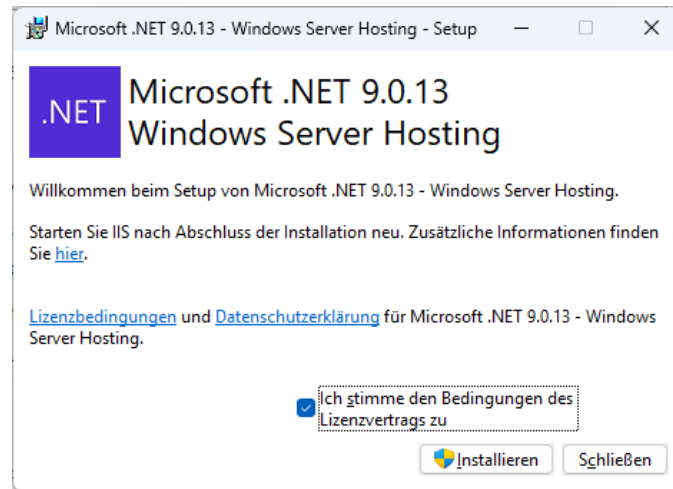


Abbildung 12: Hosting Bundle Installation

Der erfolgreiche Abschluss der Installation des Hosting Bundles wird mit dem folgenden Dialog angezeigt.

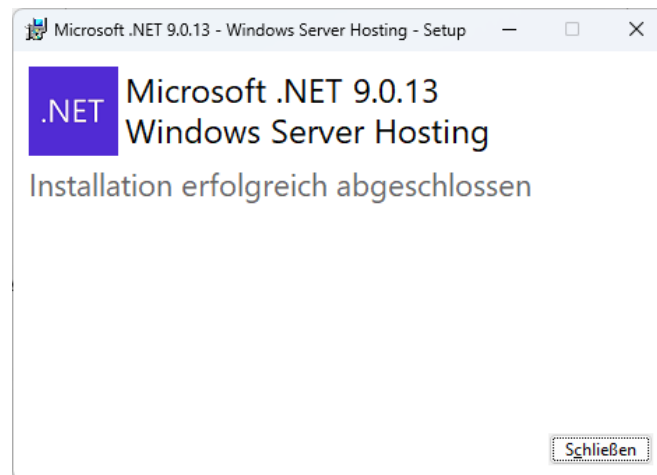
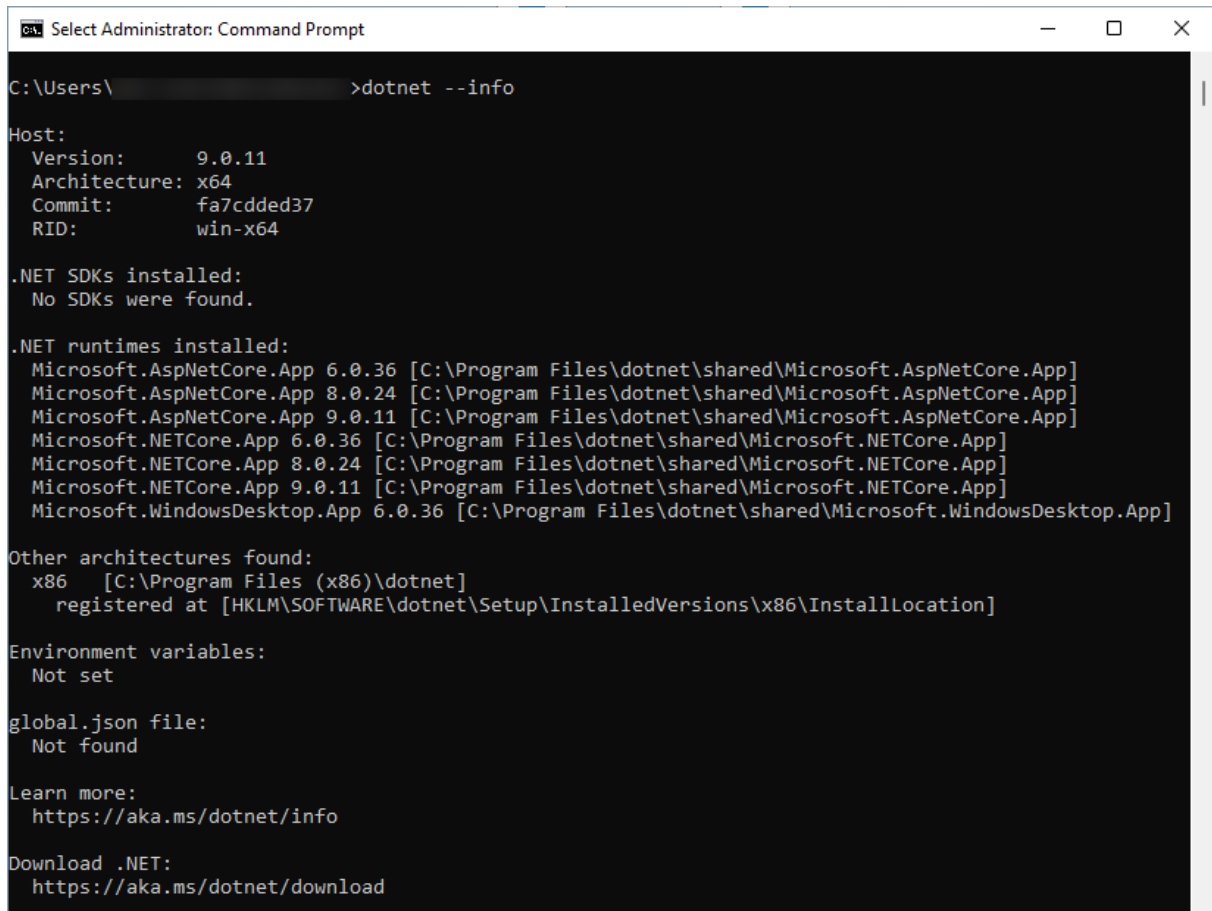


Abbildung 13: Hosting Bundle erfolgreich installiert

5.1.5 Anzeige der installierten .NET Versionen

Mit dem Befehl `dotnet --info` können die Versionsinformationen der installierten .NET-Versionen angezeigt werden.



```

C:\Users\ >dotnet --info

Host:
  Version:      9.0.11
  Architecture: x64
  Commit:       fa7cdded37
  RID:          win-x64

.NET SDKs installed:
  No SDKs were found.

.NET runtimes installed:
  Microsoft.AspNetCore.App 6.0.36 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 8.0.24 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 9.0.11 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 6.0.36 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 8.0.24 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 9.0.11 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.WindowsDesktop.App 6.0.36 [C:\Program Files\dotnet\shared\Microsoft.WindowsDesktop.App]

Other architectures found:
  x86 [C:\Program Files (x86)\dotnet]
     registered at [HKLM\SOFTWARE\dotnet\Setup\InstalledVersions\x86\InstallLocation]

Environment variables:
  Not set

global.json file:
  Not found

Learn more:
  https://aka.ms/dotnet/info

Download .NET:
  https://aka.ms/dotnet/download
  
```

Abbildung 14: Installierte .NET-Versionen

5.1.6 PIPEFAB Connect Verzeichnis anlegen

Für die Aufnahme der PIPEFAB Connect Web-API Programmdateien muss unterhalb des Pfades C:\inetpub das Verzeichnis *PIPEFAB Connect WebApi* neu angelegt werden.

Die folgende Abbildung zeigt das neu angelegte Verzeichnis unterhalb von C:\inetpub.

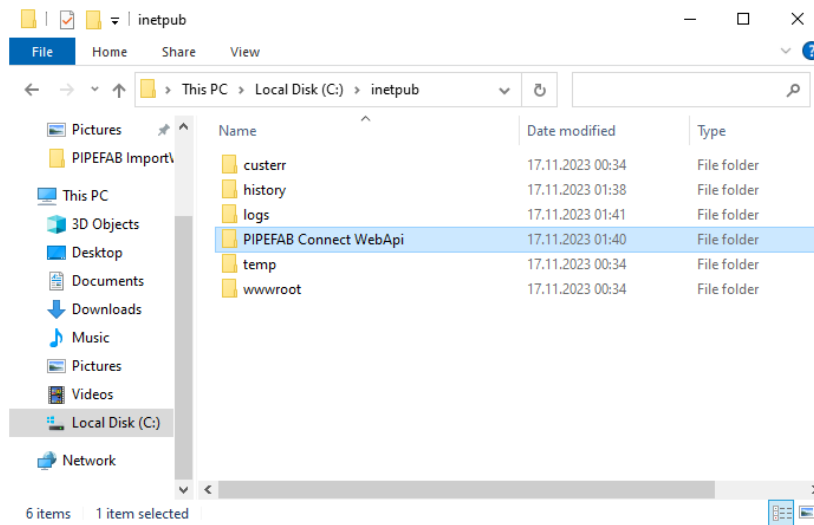


Abbildung 15: Programmverzeichnis anlegen

5.1.7 Web Server IIS für PIPEFAB Connect WebApi einrichten

Zum Einrichten des Web Servers wird der Internet Information Services (IIS) Manager verwendet. Dieser ist in den *Administrativen Tools* zu finden. Siehe folgende Abbildung.

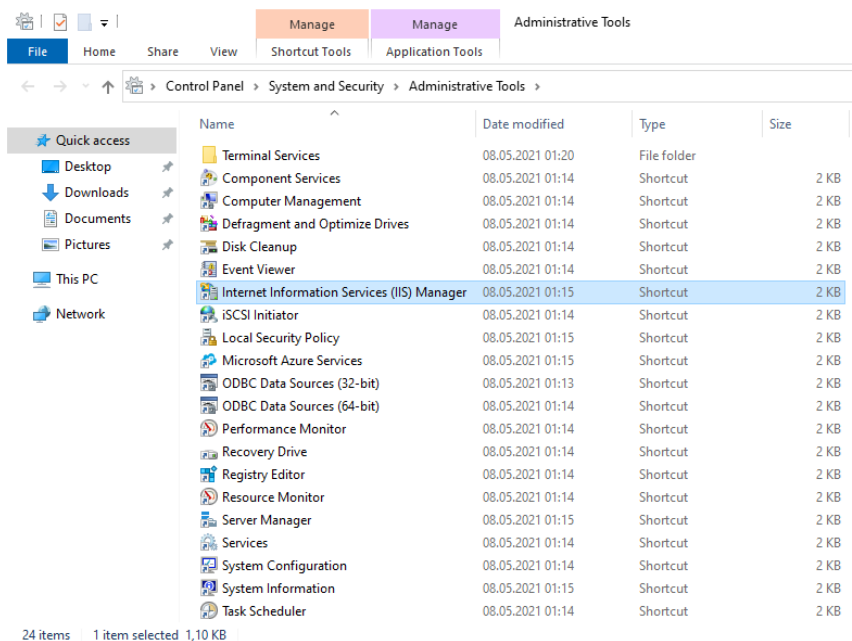


Abbildung 16: IIS-Manager starten

Nach dem Start des IIS-Managers sollte sich die folgende Ansicht im IIS-Manager erscheinen.

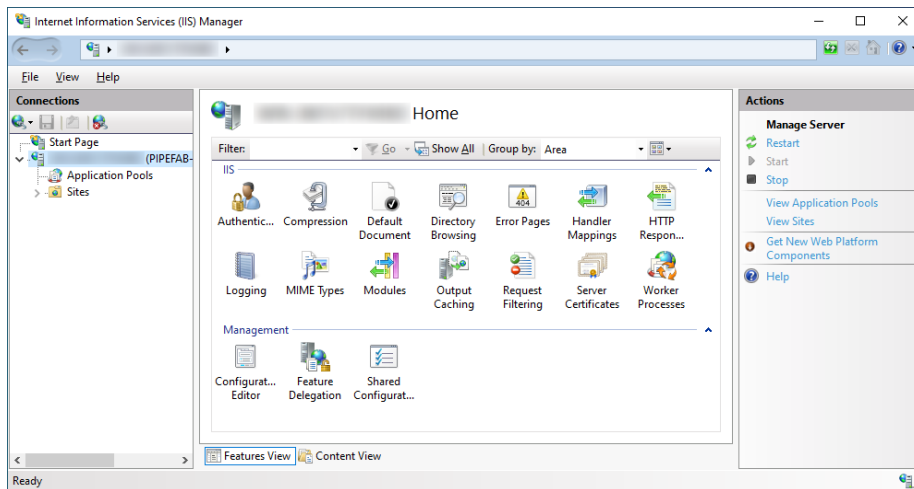


Abbildung 17: Ansicht IIS-Manager

Als erstes muss für das PIPEFAB-Connect Web-API ein neuer Application Pool angelegt werden. Die folgende Abbildung zeigt die dafür notwendigen Parameter.

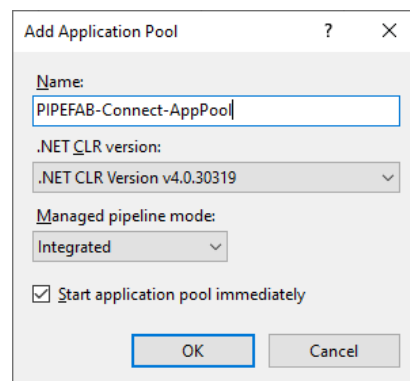


Abbildung 18: PIPEFAB-Connect AppPool

Nach dem erfolgreichen Anlegen und starten des neuen PIPEFAB Connect Application Pools sollte der neue Application Pool in der Liste der Application Pools wie folgt dargestellt werden.

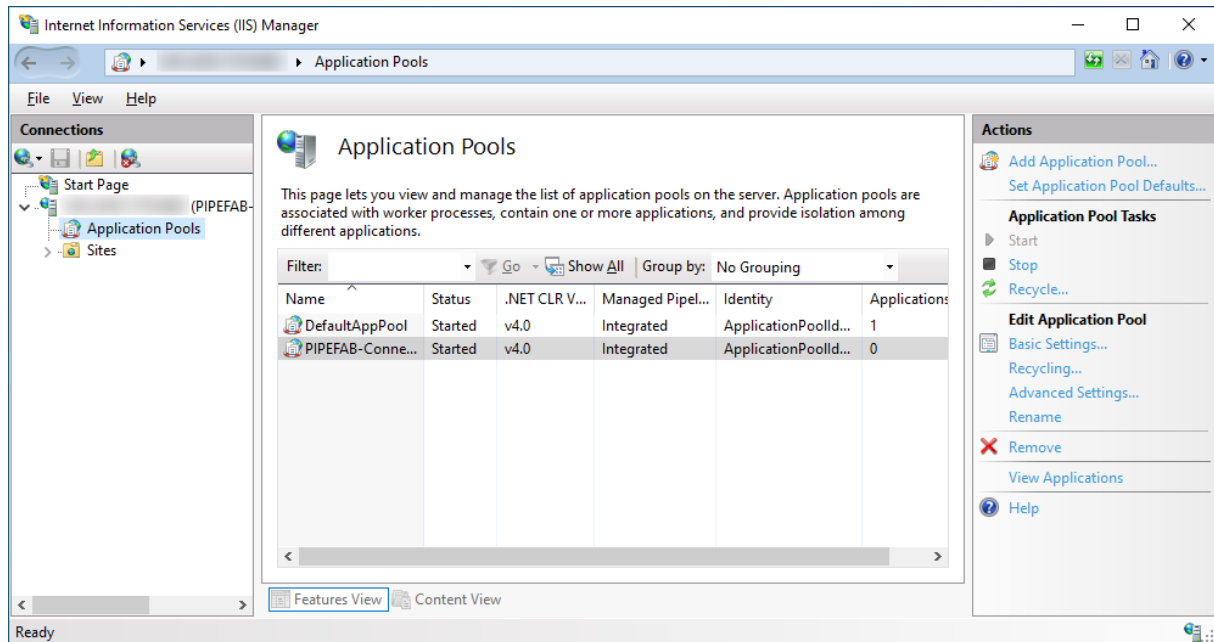


Abbildung 19: Liste der Application Pools

Als nächstes muss die IIS Default Website gestoppt werden. Dazu das Context-Menü unter Sites > Default Website öffnen und unter Manage Website die Default Website stoppen. Nach dem Stoppen der Default Website sollte die Ansicht des IIS-Managers sich wie folgt darstellen.

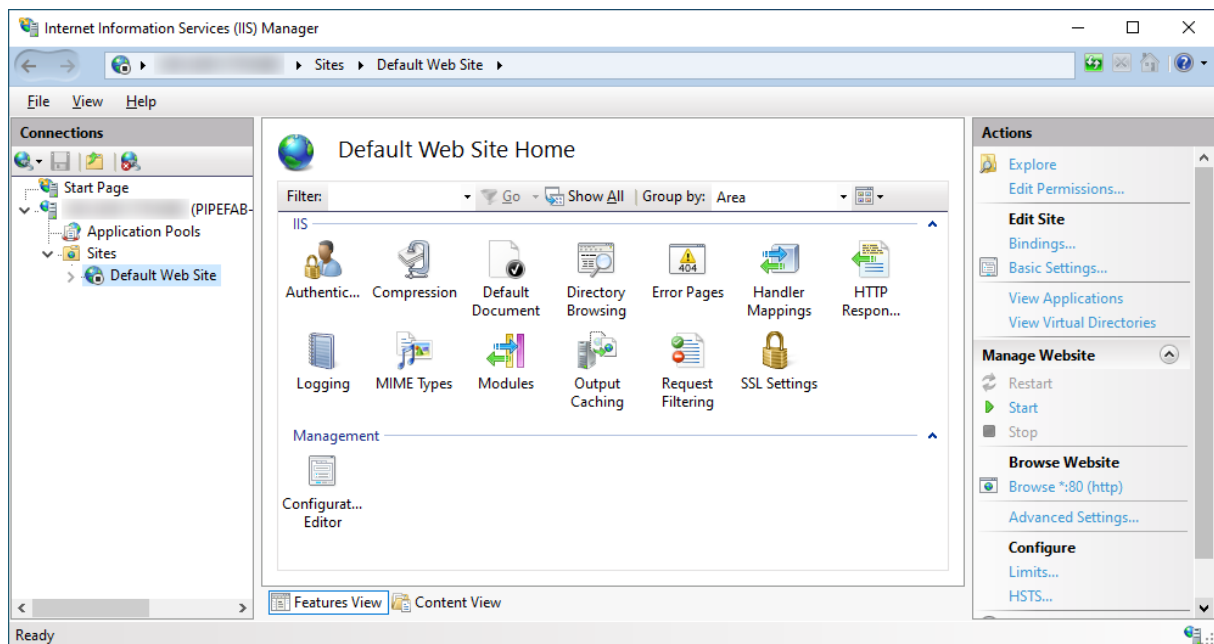


Abbildung 20: Default Website gestoppt

Über das Sites Context-Menü muss für das PIEFAB Connect Web-API eine neue Website angelegt werden. Die dazu notwendigen Einstellungen sind auf der folgenden Abbildung angegeben.

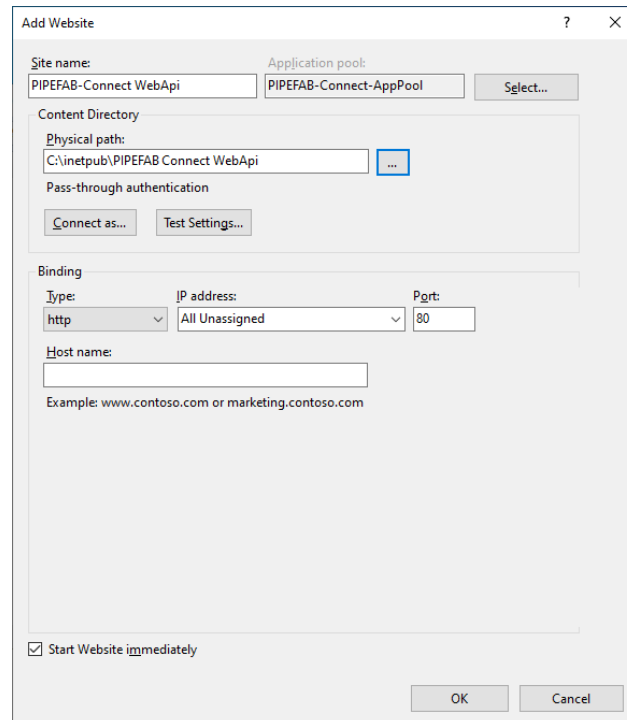
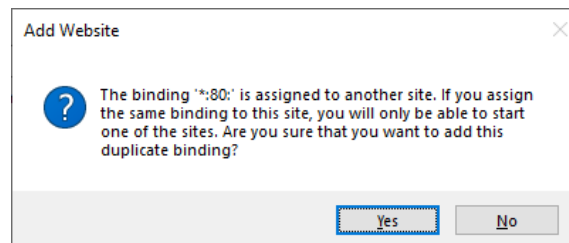


Abbildung 21: PIPEFAB Connect Website

Beim Anlegen der PIPEFAB Connect Website erscheint folgender Hinweis:



Dieser Hinweis kann einfach mit Yes bestätigt werden. Da die Default Website gestoppt ist, kann diese Meldung ignoriert werden.

Nach dem Anlegen der neuen PIPEFAB Connect Website sollte sich die IIS-Manager Ansicht wie folgt darstellen.

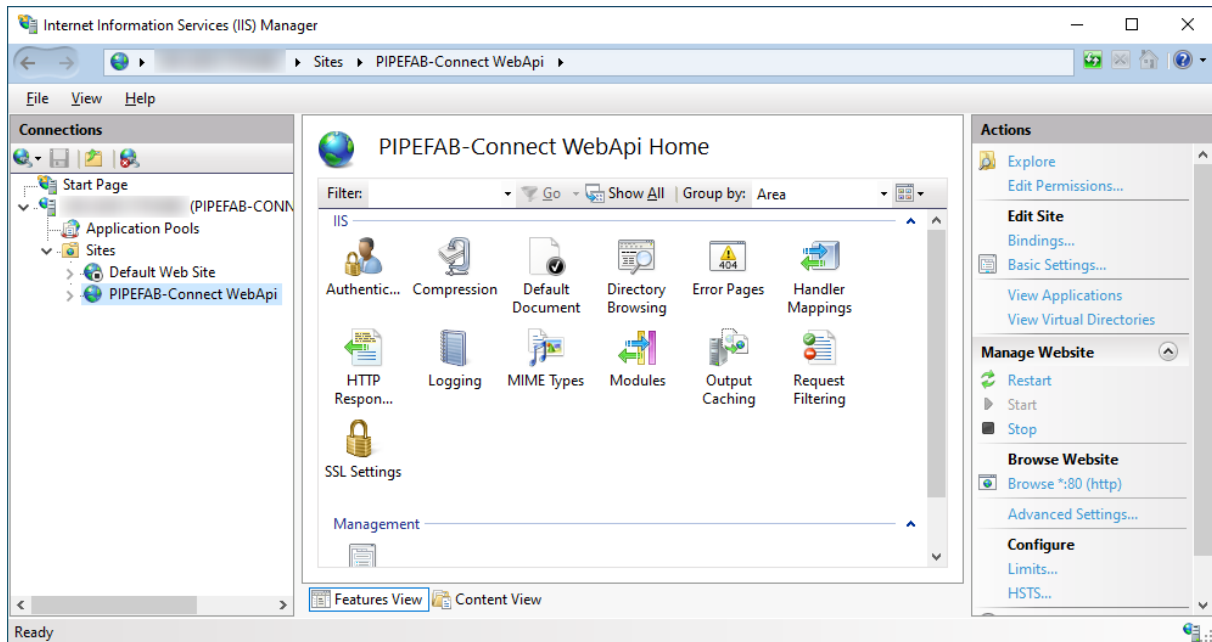


Abbildung 22: IIS-Manager Ansicht mit PIPEFAB-Connect

Damit ist die Website-Einrichtung für PIPEFAB Connect abgeschlossen. Im nächsten Schritt müssen nun die Programmdateien in das angelegte Verzeichnis kopiert werden.

5.1.8 Programmdateien kopieren

Die Programmdateien des PIPEFAB Connect Web-API's befinden sich in einem gezip'ten Archiv, welches zuerst entpackt werden muss. Nach dem Entpacken sind alle entpackten Dateien in das unter 4.1.4 angelegte PIPEFAB Connect Verzeichnis zu kopieren. Der Zugriff auf die PIPEFAB Datenbank und dem PIPEFAB Lizenzserver ist in der Datei appsettings.json zu parametrieren.

5.1.9 Datenbankzugriff und Lizenzserver in appsettings.json einstellen

Die für den Verbindungsaufbau zur Datenbank notwendigen Parameter sowie die Verbindungsinformationen zum PIPEFAB Lizenzserver werden für das PIPEFAB Connect Web-API in der Datei appsettings.json angegeben. Die Datei appsettings.json befindet sich in dem unter Kapitel 5.1.6 angelegten Programmverzeichnis des PIPEFAB Connect Web-API's.

In der folgenden Abbildung ist der schematische Aufbau der Datei appsettings.json abgebildet.

```
{
  "Url": "http://localhost:8080/",
  "JwtSettings": {....},
  "DatabaseConnection": {....},
  "PipeFabLicenseOption": {....},
  "Logging": {....},
  "NLog": {....},
  "AllowedHosts": "*"
}
```

Abbildung 23: Schematischer Aufbau der appsettings.json

In der folgenden Abbildung ist schematisch die Übernahme der Datenbankverbindungs-einstellungen aus dem PIPEFAB DatabaseConnectionInfo.xml und dem PIPEFAB Licence-ServerConnectionInfo.xml in die appsettings.json Datei des PIPEFAB-Connect Web-API's dargestellt.



Abbildung 24: Übernahme der Verbindungseinstellungen

Hinweis:

Die Angabe von `ENC%` bei den Passwörtern bedeutet, dass diese Passwörter in verschlüsselter Form vorliegen. Sollten Sie unverschlüsselte Passwörter verwenden, entfällt die Angabe von `%ENC%`.

Die Verwendung einer `appsettings.json` Datei aus dem Programmverzeichnis einer vorherigen PIPEFAB-Connect Web-API Installation führt dazu, dass das Web-API nicht korrekt startet. Die `appsettings.json` Datei einer Vorgängerversion muss an das aktuell gültige Format angepasst werden.

5.1.10 HTTPS-Zugriff einrichten

Das Einrichten von HTTPS erfordert die Installation eines SSL-Zertifikats auf dem IIS. Nach der Installation des SSL-Zertifikats muss zudem das Binding der Website auf HTTPS umgestellt werden. Das Http-Binding zur PIPEFAB-Connect Website ist dann zu entfernen.

5.1.11 Funktionstest

Der Funktionstest kann mit Hilfe eines Internet-Browsers durchgeführt werden. Unter der Url [Server]/api/state meldet das PIPEFAB-Connect Web-API den aktuellen Status (siehe folgende Abbildung).

```
{
  "state": "OK",
  "version": "4.7.62.14860",
  "dbVersion": "860",
  "licenses": [
    "PIPEFAB Connect WebAPI PCF-Import",
    "PIPEFAB Connect WebAPI Kataloge",
    "PIPEFAB Connect WebAPI Projektkatalog"
  ],
  "webApi": "2"
}
```

Abbildung 25: Status des PIPEFAB-Connect WebApi's

5.2 PIPEFAB Connect ImportWorker Service

Die an das PIPEFAB Connect Web-API übermittelten Aufträge (Jobs) werden von dem PIPEFAB Connect ImportWorker Service übernommen und bearbeitet. Je Windowsserver kann max. ein PIPEFAB Connect ImportWorker Service installiert werden. Diese Kapitel befasst sich mit der Installation des PIPEFAB Connect ImportWorker Services.

5.2.1 Systemvoraussetzungen

Diese Installationsanleitung basiert auf der Annahme, dass der PIPEFAB Connect ImportWorker Service auf einem Windows Server 2022 Standard (Desktop Experience) mit voreingestellter englischer Sprache installiert wird. Die folgende Abbildung zeigt die Versionsinformation des hier verwendeten Windows Server 2022.

About

Windows specifications

Edition	Windows Server 2022 Standard
Version	21H2
Installed on	16.11.2023
OS build	20348.2113

Abbildung 26: Versionsinformation

Bevor mit den vorbereitenden Maßnahmen zur Installation des PIPEFAB Connect ImportWorker Services begonnen wird, sollten alle notwendigen Microsoft Sicherheitsupdates auf dem Server installiert sein. Diese gilt auch für zukünftige Microsoft Sicherheitsupdates.

Der PIPEFAB Connect ImportWorker Service und das PIPEFAB Connect Web-API sollten auf getrennten Windows Servern installiert werden um Performance- und Sicherheitsproblemen zu vermeiden.

5.2.2 Lizenzierung

Zum Betreiben des PIPEFAB Connect ImportWorker Service wird das Lizenzpaket *PIEFAB 4 Connect SMAP 3D* benötigt. In diesem Lizenzpaket ist die benötigte Lizenzen *PIEFAB Connect ImportWorker* enthalten.

5.2.3 PIPEFAB Connect Verzeichnis anlegen

Für die Aufnahme der PIPEFAB Connect ImportWorker Service Programmdateien muss unterhalb des Pfades C:\Program Files das Verzeichnis „PIPEFAB Connect ImportWorkerService“ neu angelegt werden. Die folgende Abbildung zeigt das angelegte Verzeichnis unter C:\Program Files.

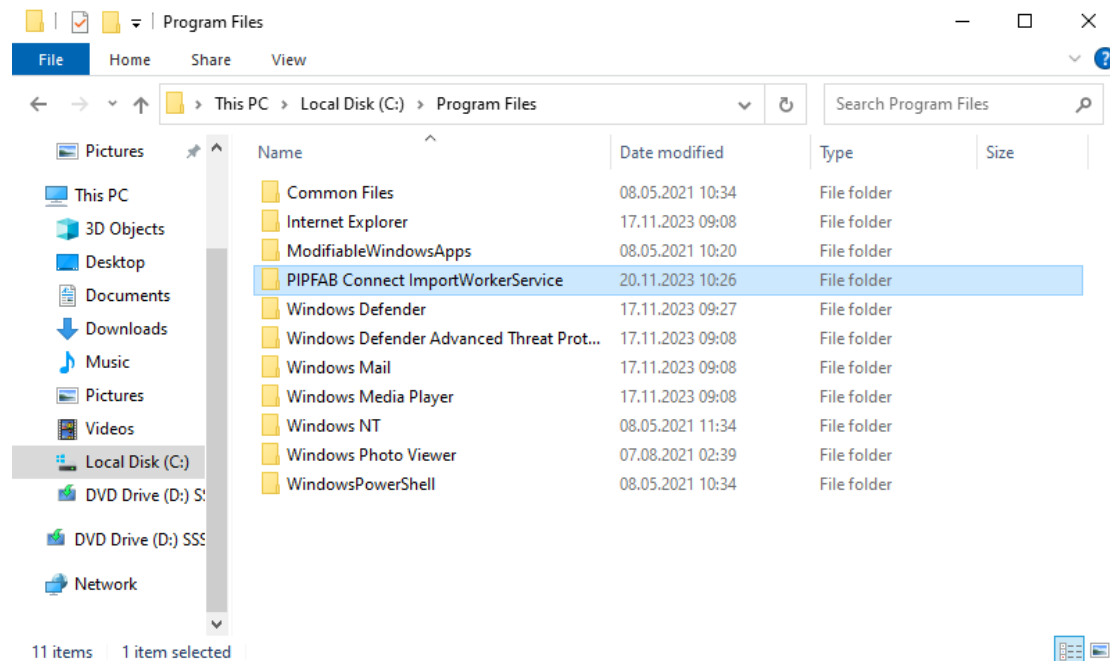


Abbildung 27: PIPEFAB Connect ImportWorker Service Verzeichnis

5.2.4 Programmdateien kopieren

Die Programmdateien des PIPEFAB Connect ImportWorker Services befinden sich in einem gezip'ten Archiv, welches zuerst entpackt werden muss. Nach dem Entpacken sind alle entpackten Dateien in das unter 5.2.3 angelegte PIPEFAB Connect ImportWorker Service Verzeichnis zu kopieren. Der Zugriff zur PIPEFAB Datenbank ist in der Datei *DatabaseConnectionInfo.xml* zu parametrieren.

Neue Versionen / Updates des PIPEFAB Connect ImportWorker Services werden analog wie oben beschrieben eingespielt.

5.2.5 Datenbankverbindung konfigurieren

Für den Zugriff auf die PIPEFAB Datenbank wird eine XML-Datei mit dem Namen DatabaseServerConnectionInfo.xml benötigt. Siehe PIPEFAB-Dokumentation.

5.2.6 Verbindung zum Lizenzserver konfigurieren

Für den Zugriff auf den PIPEFAB-Lizenzserver wird eine XML-Datei mit dem Namen LicenceServConnectionInfo.xml benötigt. Siehe PIPEFAB-Dokumentation.

5.2.7 PIPEFAB Connect ImportWorker konfigurieren

Der PIPEFAB Connect ImportWorker Service kann für unterschiedliche Einsatzszenarien (z.B. ein Worker, mehrere Worker, WorkerPool etc.) konfiguriert werden.

Die notwendigen Angaben werden in der Datei

TT.PipeFab.Connect.ImportWorker.Services.exe.config, (die sich Programmverzeichnis des PIPEFAB Connect ImportWorker Services befindet) vorgenommen.

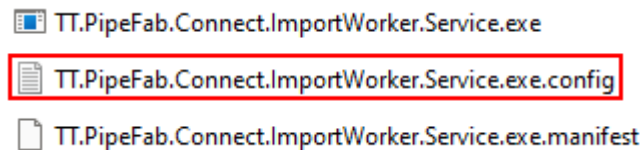


Abbildung 28: Konfigurationsdatei

In der Konfigurationdatei können folgende Angaben angegeben werden:

a) WorkerNumber

Jeder Worker muss über eine eindeutige Nummerierung verfügen. Es wird empfohlen jedem Worker eine fortlaufende Nummer beginnend bei 1 zuzuordnen. Unabhängig von der Worker-Pool Zuordnung.

Wertebereich: Numerisch.

Beispiel:

```
<add key="WorkerNumber" value="1"/>
```

b) WorkerPool

Jeder Worker kann einem oder mehreren sog. Worker-Pool's zugeordnet werden. Die zu einem Worker-Pool zugeordneten Worker verarbeiten nur die Aufträge die Ihrem Worker-Pool zugeordnet sind. Siehe hierzu das Kapitel „Konfiguration des Worker-Pool's“.

Wird kein Worker-Pool in der Konfiguration angegeben, werden nur Aufträge verarbeitet die keinem Worker-Pool zugeordnet sind.

Wertebereich: Text, max. 60 Zeichen

Beispiel (kein Worker-Pool zugeordnet):

```
<add key="WorkerPool" value=""/>
```

Beispiel (Zuordnung zu Worker-Pool „CadPool“):

```
<add key="WorkerPool" value="CadPool"/>
```

Beispiel:

c) WorkerCycleTime

Mit dem Wert der WorkerCycleTime wird festgelegt, wie lange der Worker wartet bis er wieder nach neuen Aufträgen in der Datenbank sucht. Nach der Abarbeitung eines Auftrags sucht der Worker ohne Wartezeit nach neuen Aufträgen. Erst wenn er keine neuen Aufträge gefunden hat, geht er für die festgelegte Zeit in einen sog. Ruhezustand. Nach Ablauf der in WorkerCycleTime festgelegten Zeit (in Millisekunden) wird die Suche nach neuen Aufträgen fortgesetzt.

Wartezeiten < 5000 Millisekunden = 5 Sekunden sollten wegen der Prozessdauer beim Suchen nach neuen Aufträgen vermieden werden.

Wertebereich: Numerisch (Angabe in Millisekunden)

Beispiel (10 Sekunden Wartezeit):

```
<add key="WorkerCycleTime" value="10000"/>
```

d) BatchSize

Die BatchSize legt die Anzahl der max. zu verarbeitenden Aufträge pro Suchvorgang und Worker fest. In einer Umgebung mit mehreren Workern, sollte der Wert für die BatchSize kleiner gewählt werden (1-5) um eventuell vorhandene Aufträge von freien Worker verarbeiten zu lassen. Bei wiederkehrenden, bekannten Auftragsgrößen sollte die BatchSize dementsprechend angepasst werden, um die Anzahl der Suchvorgänge nach neuen Aufträgen zu reduzieren (BatchSize = Anzahl Aufträge / Anzahl Worker)

Wertebereich: Numerisch, 1-255

Beispiel (BatchSize = 5):

```
<add key="BatchSize" value="5"/>
```

Die folgende Abbildung zeigt den Inhalt (Auszug) einer Worker Konfigurationsdatei.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <appSettings>
    <add key="WorkerNumber" value="1"/>
    <add key="WorkerPool" value="" />
    <add key="WorkerCycleTime" value="10000"/>
    <add key="BatchSize" value="5"/>
  </appSettings>
```

Abbildung 29: Worker-Konfigurationsdatei (Auszug)

5.2.8 Konfiguration des Worker-Pool's

Jeder Worker kann einem oder mehreren sog. Worker-Pool's zugeordnet werden. Die zu einem Worker-Pool zugeordneten Worker verarbeiten nur die Aufträge die Ihrem Worker-Pool zugeordnet worden sind.

Wenn sich ein Worker im Leerlauf befindet, kann er Aufträge aus anderen Worker-Pool's verarbeiten. Dazu ist in der Konfiguration eine Liste mit den Namen der Worker-Pool's anzugeben (durch Kommata getrennt) aus denen Aufträge bei Leerlauf bearbeitet werden sollen. Der erste Name in der Liste Worker-Pool's ist dabei der priorisierte Worker-Pool. Aufträge aus den anderen konfigurierten Worker-Pools werden erst dann verarbeitet, wenn keine Aufträge für den priorisierten Worker-pool vorliegen.

Beispiel:

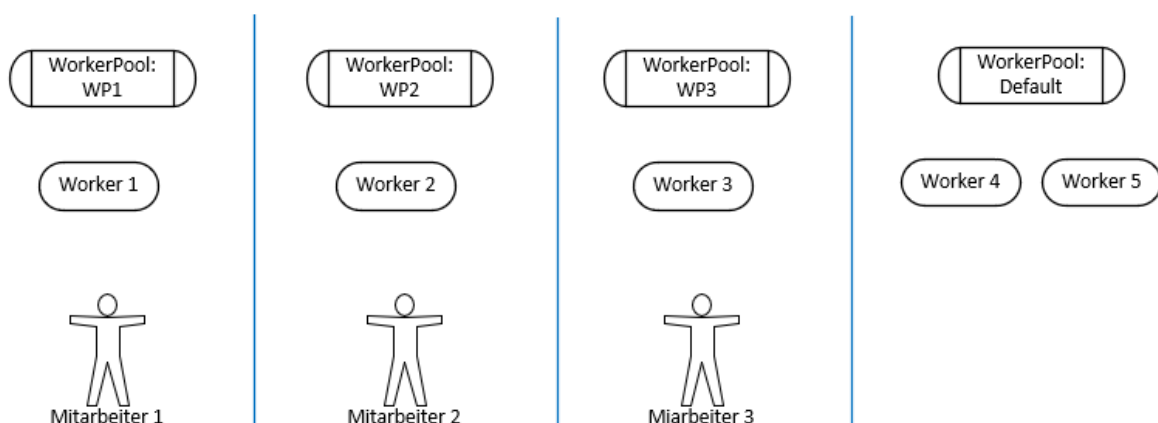
Für drei Mitarbeiter ist je ein PIPEFAB Connect ImportWorker auf einem Server installiert worden. Zusätzlich sind noch zwei Server mit je einem ImportWorker für den WorkerPool „Default“ installiert worden. Die Zuweisung der ImportWorker wird anhand des Worker-Pools wie folgt vorgenommen:

Mitarbeiter 1 -> WorkerPool: WP1

Mitarbeiter 2 -> WorkerPool: WP2

Mitarbeiter 3 -> WorkerPool: WP3

Zusätzlich sollen alle ImportWorker wenn diese sich im Leerlauf befinden, Aufträge der anderen Worker-Pools bearbeiten um diese zu unterstützen. Leerlauf bedeutet dabei, dass keine Aufträge für ihren priorisierten WorkerPool vorhanden sind.



Die folgenden Abbildungen zeigen die Konfiguration der einzelnen ImportWorker.

Konfiguration für Worker 1

WP1	WP2	WP3
-----	-----	-----

Prio

Prüfreihefolge (Leerlauf): WP1 -> WP2 -> WP1 -> WP3

Konfiguration für Worker 2

WP2	WP3	WP1
-----	-----	-----

Prio

Prüfreihefolge (Leerlauf): WP2 -> WP3 -> WP2 -> WP1

Konfiguration für Worker 3

WP3	WP1	WP2
-----	-----	-----

Prio

Prüfreihefolge (Leerlauf): WP3 -> WP1 -> WP3 -> WP2

```
<appSettings>
  <add key="WorkerNumber" value="1" />
  <add key="WorkerPool" value="WP1,WP2,WP3" />
  <add key="WorkerCycleTime" value="10000" />
  <add key="BatchSize" value="5" />
  <add key="CleanUpThreshold" value="500" />
</appSettings>
```

```
<appSettings>
  <add key="WorkerNumber" value="2" />
  <add key="WorkerPool" value="WP2,WP3,WP1" />
  <add key="WorkerCycleTime" value="10000" />
  <add key="BatchSize" value="5" />
  <add key="CleanUpThreshold" value="500" />
</appSettings>
```

```
<appSettings>
  <add key="WorkerNumber" value="3" />
  <add key="WorkerPool" value="WP3,WP1,WP2" />
  <add key="WorkerCycleTime" value="10000" />
  <add key="BatchSize" value="5" />
  <add key="CleanUpThreshold" value="500" />
</appSettings>
```

Konfiguration für Worker 4

Default	WP1	WP2	WP3
---------	-----	-----	-----

Prio

Prüfreihenfolge (Leerlauf): Default -> WP1 -> Default -> WP2 -> Default -> WP3

Konfiguration für Worker 5

Default	WP2	WP3	WP1
---------	-----	-----	-----

Prio

Prüfreihenfolge (Leerlauf): Default -> WP2 -> Default -> WP3 -> Default -> WP1

```
<appSettings>
  <add key="WorkerNumber" value="4" />
  <add key="WorkerPool" value="Default,WP1,WP2,WP3" />
  <add key="WorkerCycleTime" value="10000" />
  <add key="BatchSize" value="5" />
  <add key="CleanUpThreshold" value="500" />
</appSettings>

<appSettings>
  <add key="WorkerNumber" value="5" />
  <add key="WorkerPool" value="Default,WP2,WP3,WP1" />
  <add key="WorkerCycleTime" value="10000" />
  <add key="BatchSize" value="5" />
  <add key="CleanUpThreshold" value="500" />
</appSettings>
```

5.2.9 PIPEFAB Connect ImportWorker Service installieren

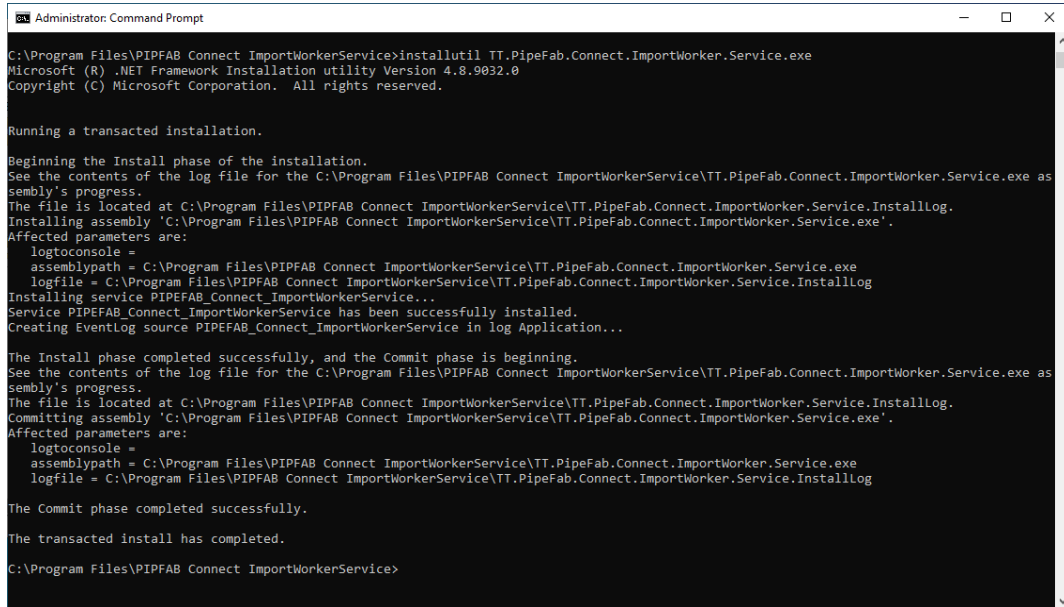
Der PIPEFAB Connect ImportWorker Service wird als Windows-Service installiert.

Dazu kann die im Programmverzeichnis vorhandene Batch-Datei „Install PIPEFAB Connect ImportWorkerService.bat“ mit Administratorrechten ausgeführt werden.

Die Batch-Datei führt das folgende Kommando aus.

installutil TT.PipeFab.Connect.ImportWorker.Service.exe

Nach erfolgreicher Ausführung des Kommandos wird folgende Ausgabe im Command-Prompt ausgegeben:



```
Administrator: Command Prompt

C:\Program Files\PIPEFAB Connect ImportWorkerService>installutil TT.PipeFab.Connect.ImportWorker.Service.exe
Microsoft (R) .NET Framework Installation utility Version 4.8.9032.0
Copyright (C) Microsoft Corporation. All rights reserved.

Running a transacted installation.

Beginning the Install phase of the installation.
See the contents of the log file for the C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.exe as
sembly's progress.
The file is located at C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.Installlog.
Installing assembly 'C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.exe'.
Affected parameters are:
  logtoconsole =
  assemblypath = C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.exe
  logfile = C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.Installlog
Installing service PIPEFAB_Connect_ImportWorkerService...
Service PIPEFAB_Connect_ImportWorkerService has been successfully installed.
Creating Eventlog source PIPEFAB_Connect_ImportWorkerService in log Application...

The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.exe as
sembly's progress.
The file is located at C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.Installlog.
Committing assembly 'C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.exe'.
Affected parameters are:
  logtoconsole =
  assemblypath = C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.exe
  logfile = C:\Program Files\PIPEFAB Connect ImportWorkerService\TT.PipeFab.Connect.ImportWorker.Service.Installlog

The Commit phase completed successfully.

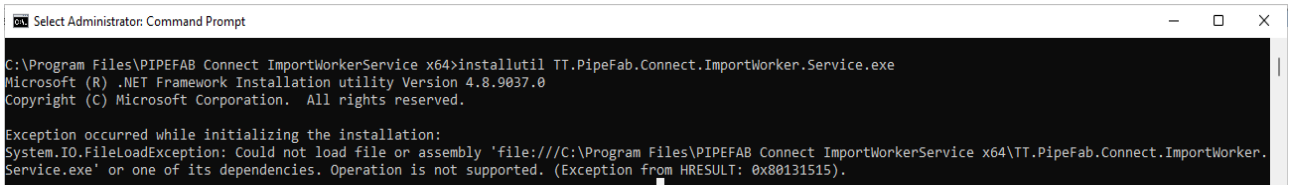
The transacted install has completed.

C:\Program Files\PIPEFAB Connect ImportWorkerService>
```

Abbildung 30: Installation Windows Service

Sollte es während der Installation zu einem Fehler (HRESULT: 0x80131515, siehe folgende Abbildung) kommen, liegt dieses möglicherweise an einer vom Windows-System eingerichteten Sperre für Programme die von einem anderen System stammen.

Fehler:



```
Select Administrator: Command Prompt

C:\Program Files\PIPEFAB Connect ImportWorkerService x64>installutil TT.PipeFab.Connect.ImportWorker.Service.exe
Microsoft (R) .NET Framework Installation utility Version 4.8.9037.0
Copyright (C) Microsoft Corporation. All rights reserved.

Exception occurred while initializing the installation:
System.IO.FileLoadException: Could not load file or assembly 'file:///C:\Program Files\PIPEFAB Connect ImportWorkerService x64\TT.PipeFab.Connect.ImportWorker.
Service.exe' or one of its dependencies. Operation is not supported. (Exception from HRESULT: 0x80131515).
```

Abbildung 31: Unblock Installationsfehler

Diese Sperre kann für die Anwendung TT.PipeFab.Connect.Import.Service.exe über die Windows-Explorer-Eigenschaften-Seite -> Unblock entfernt werden. Sie folgende Abbildung zeigen diese Seite inkl. der Unblock-Auswahlmöglichkeit.

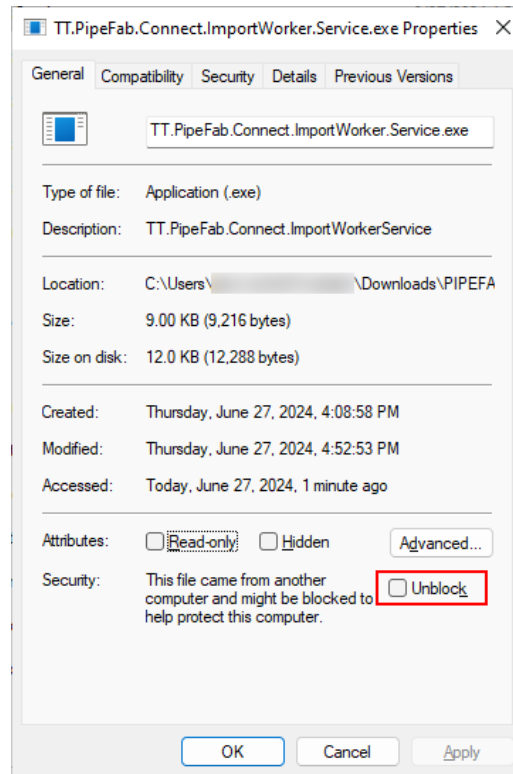


Abbildung 32: Eigenschaften des TT.PIPEFAB.Connect.Import.Services inkl. Unblock- Möglichkeit

Im Windows Dienst-Manager ist der eben installierte PIPEFAB Connect ImportWorker Service wie in der folgenden Abbildung dargestellt zu finden.

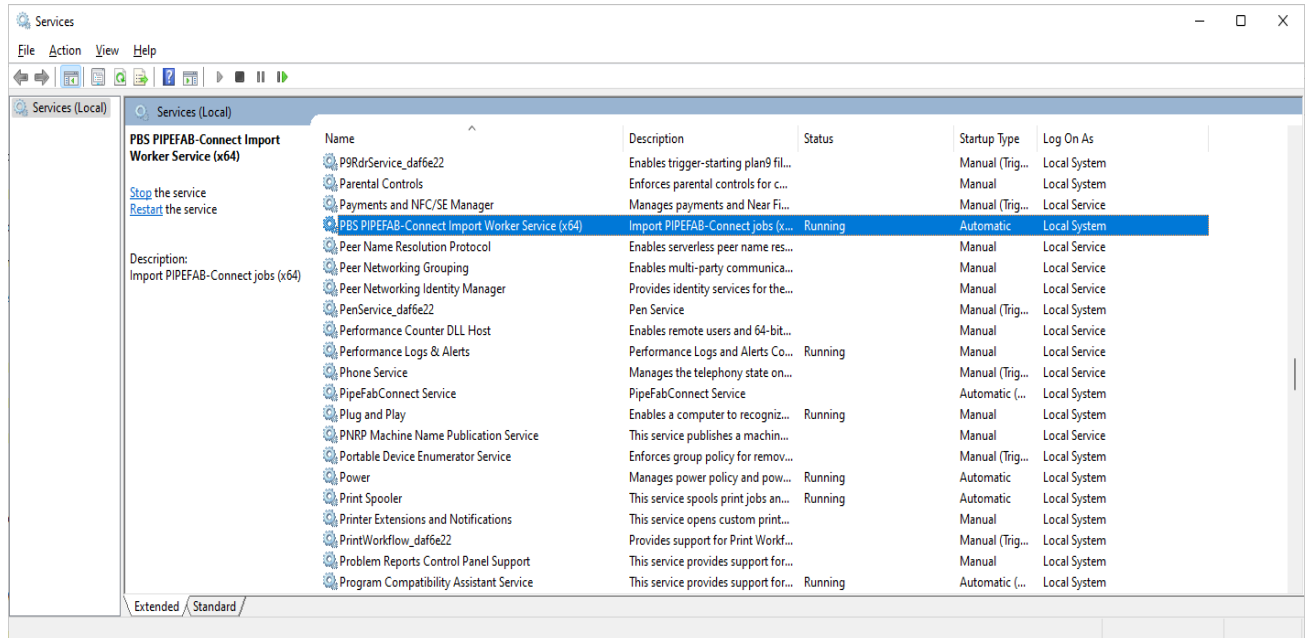


Abbildung 33: PIPEFAB Connect ImportWorker Service x64

Der installierte PIPEFAB Connect ImportWorker Service muss noch für den Betrieb konfiguriert werden. Dazu wird ein lokales User-Konto benötigt, mit dessen Berechtigungen der PIPEFAB Connect ImportWorker Service auf dem lokalen System läuft. Ein entsprechender User ist ggf. anzulegen.

Zum Konfigurieren des PIPEFAB Connect ImportWorker Services sind die Eigenschaften des Dienstes aufzurufen.

Auf der ersten Seite der Eigenschaften des Services sollten die in den folgenden abgebildeten Einstellungen gesetzt sein.

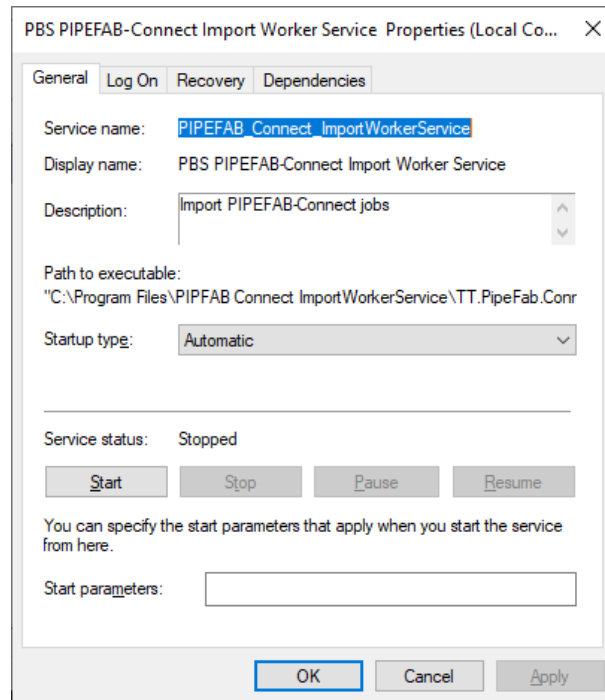


Abbildung 34: Service Eigenschaften, Allgemein

Auf dem Tab-Reiter *Log On* ist ein User-Konto anzugeben unter dessen Kontext der PIPEFAB Connect ImportWorker Service läuft (mit dessen Berechtigungen). Es werden u.a. Berechtigung benötigt, um in das Filesystem zu schreiben.

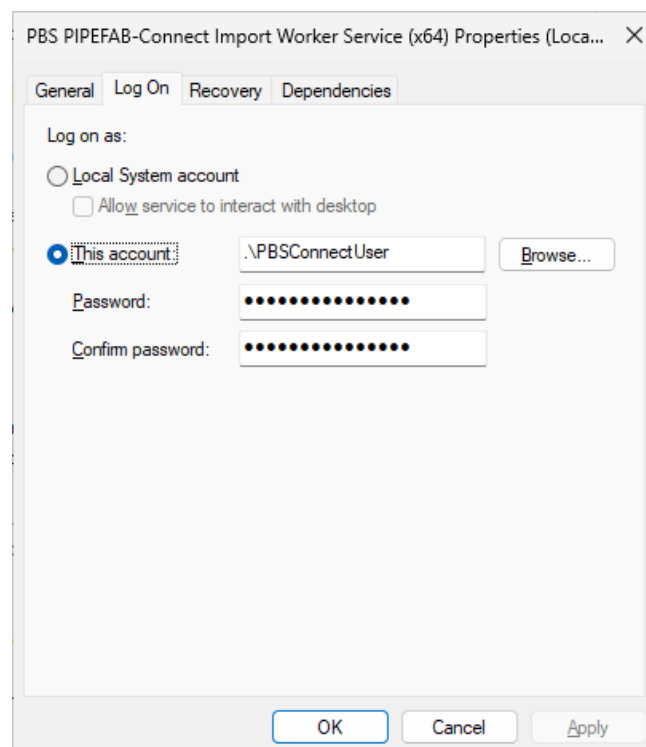


Abbildung 35: Service Eigenschaften, Anmeldung

Auf dem Tab-Reiter Recovery sind die Einstellungen gemäß der folgenden Abbildung einzustellen.

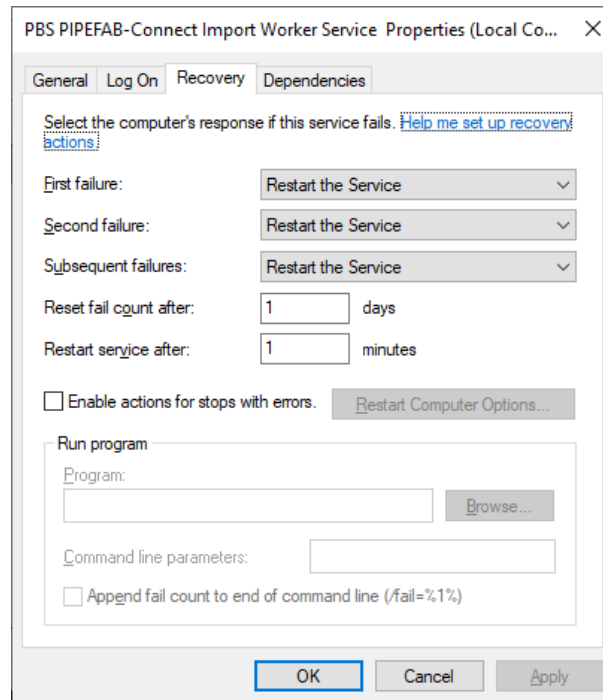


Abbildung 36: Service Eigenschaften, Recovery

Damit der User unter dessen Berechtigungen der PIPEFAB Connect ImportWorker Service ausgeführt wird, diesen Service auch Beenden und Starten darf, müssen diesem User die entsprechenden Rechte eingeräumt werden. Bitte verwenden Sie ein Tool Ihrer Wahl um die Service-Benutzerberechtigungen zu setzen. Hinweise finden Sie u.a. hier: <https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/grant-users-rights-manage-services>

1. Möglichkeit: Verwendung der Microsoft Management Console (mmc.exe)
 - Starten Sie die mmc.exe als Administrator
 - Wählen Sie unter File -> Add/Remove Snap-in...
 - Wählen Sie die in der folgenden Abbildung dargestellten Snap-Ins aus

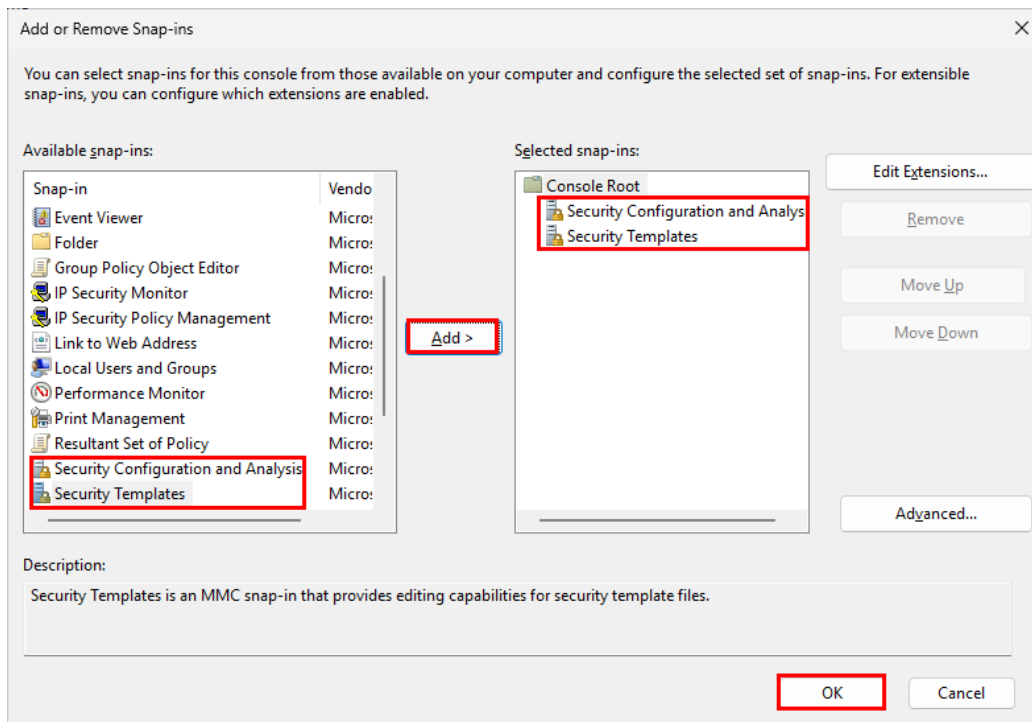


Abbildung 37: Benötigte Snap-Ins

- Anlage eines neuen Templates durch die in den folgenden abgebildeten Schritten

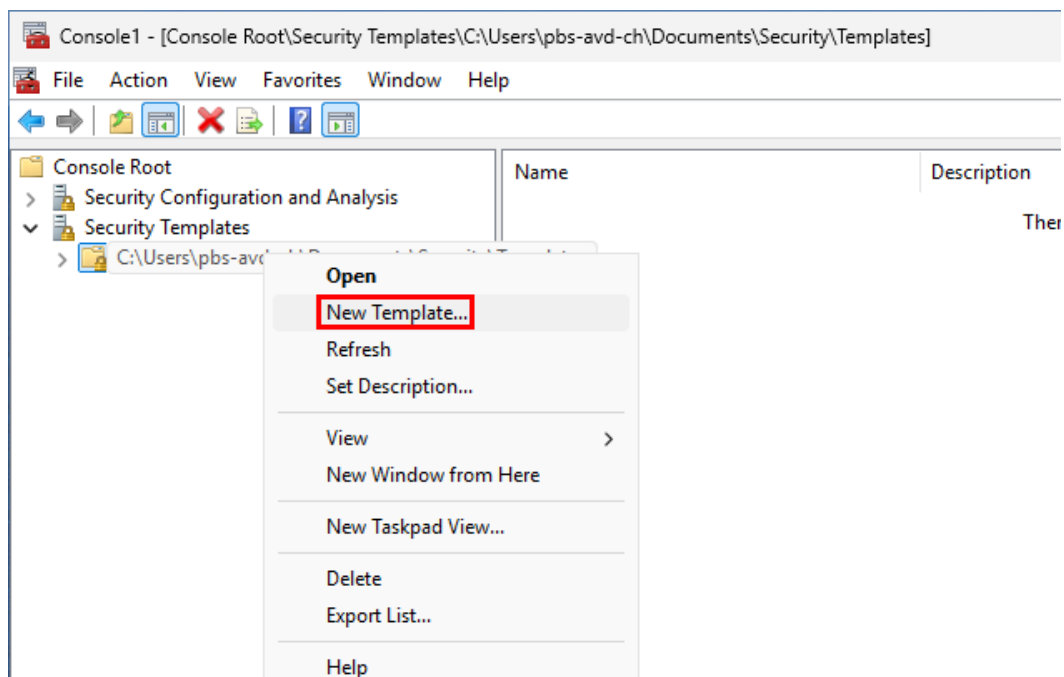


Abbildung 38: Neues Security Template anlegen

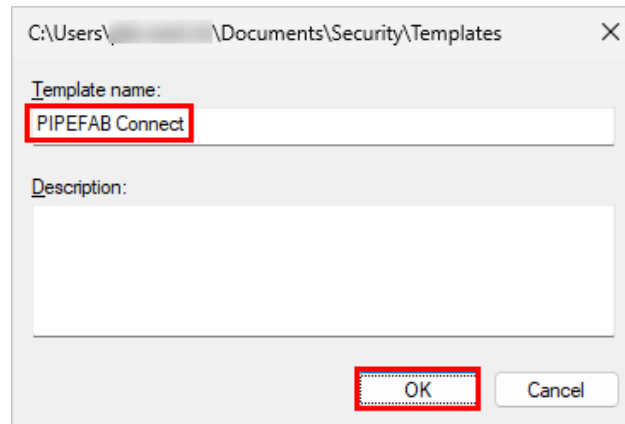


Abbildung 39: Name des neuen Templates

- Anlage einer neuen Security Datenbank durch die in den folgenden abgebildeten Schritten

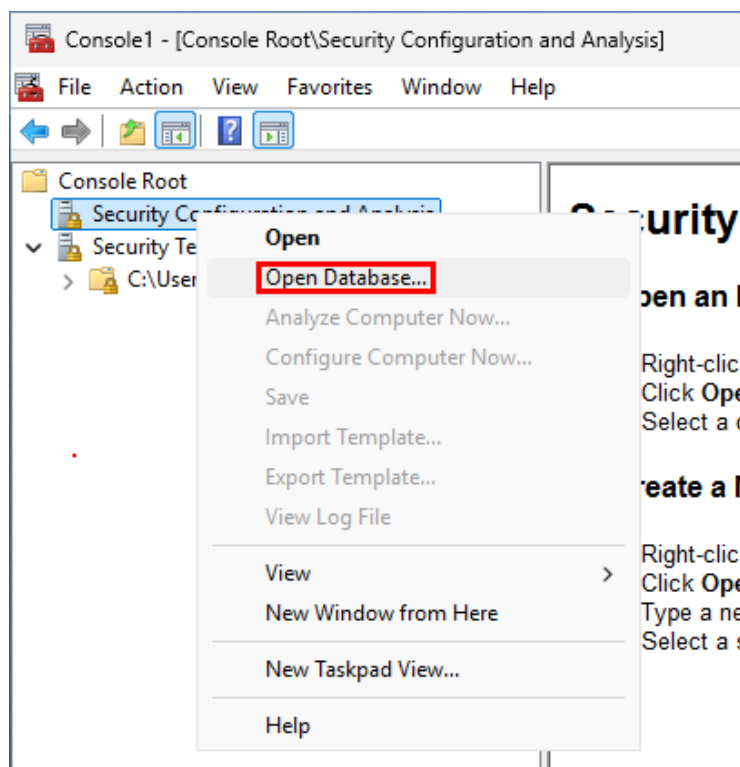


Abbildung 40: Security-DB öffnen bzw. anlegen

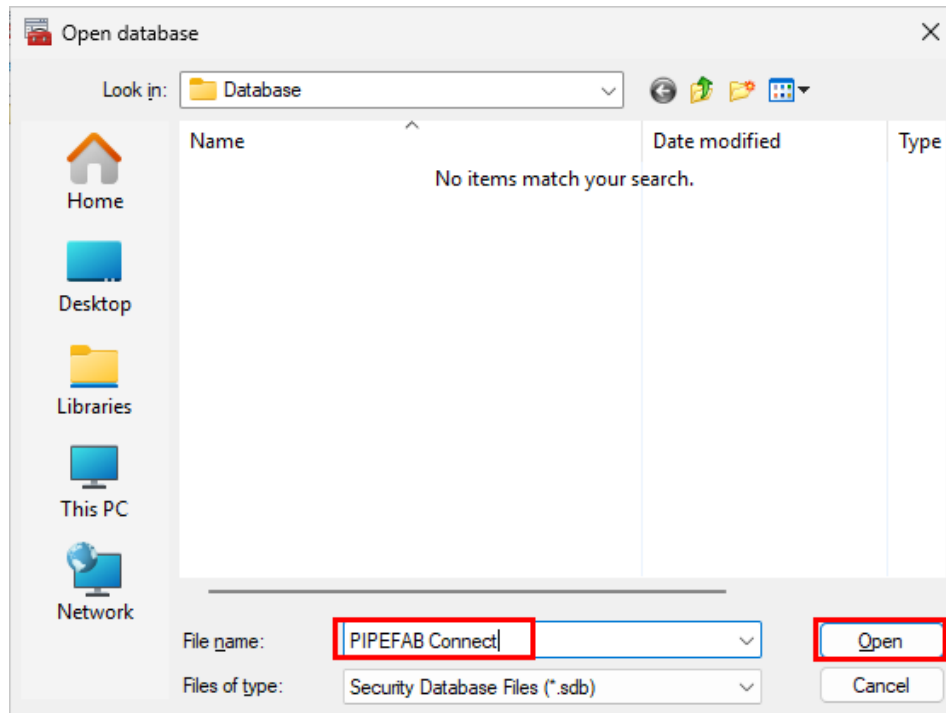


Abbildung 41: Name der Datenbank vergeben

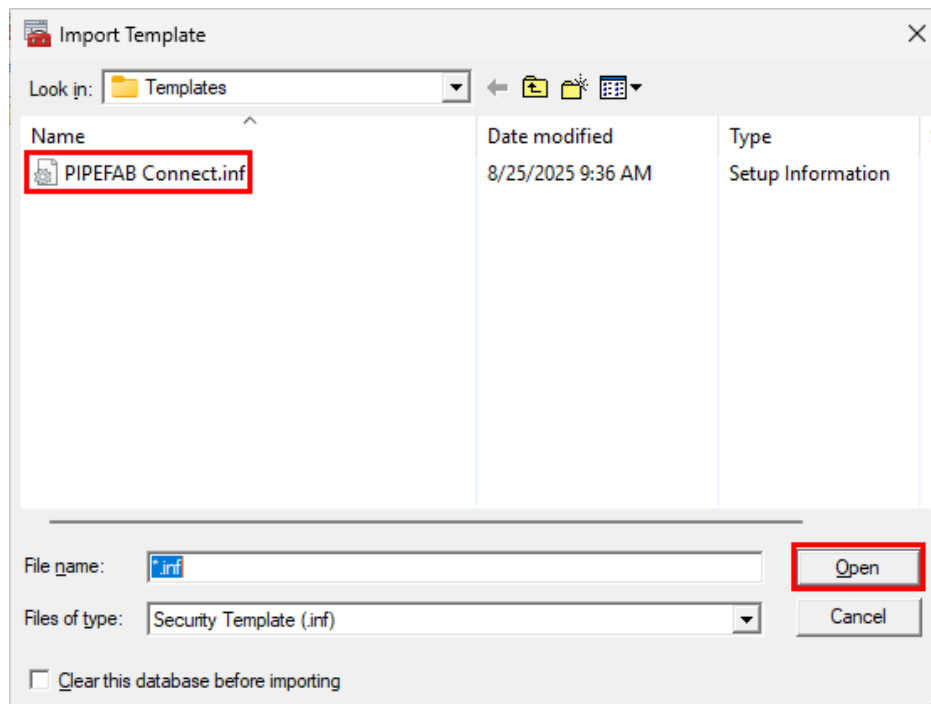


Abbildung 42: Security Template auswählen

- Als nächstes muss eine Computer Analyse ausgeführt werden.

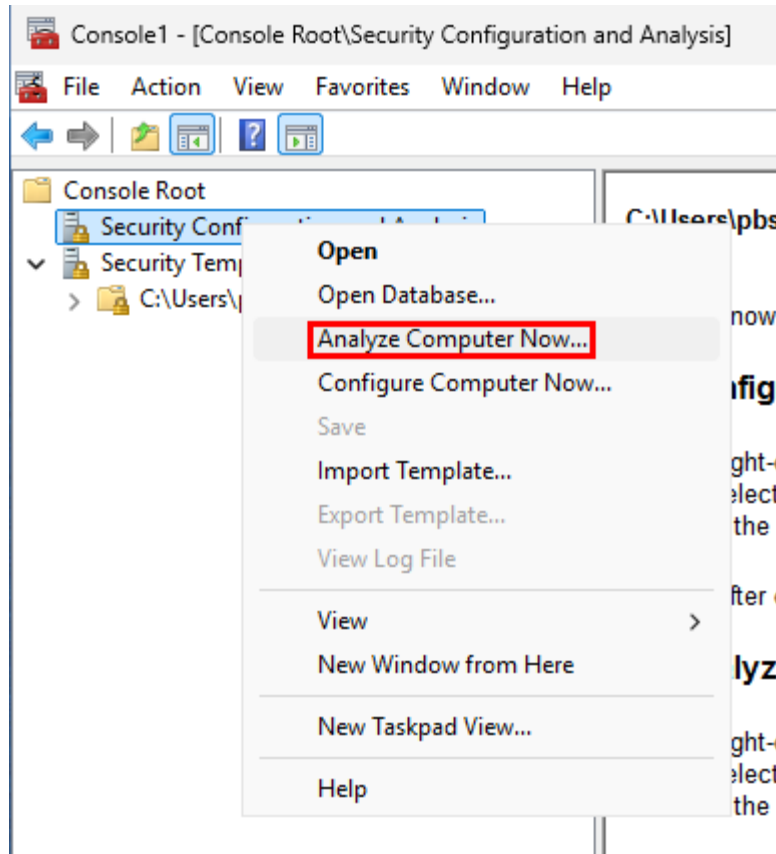


Abbildung 43: Computer Analyse durchführen

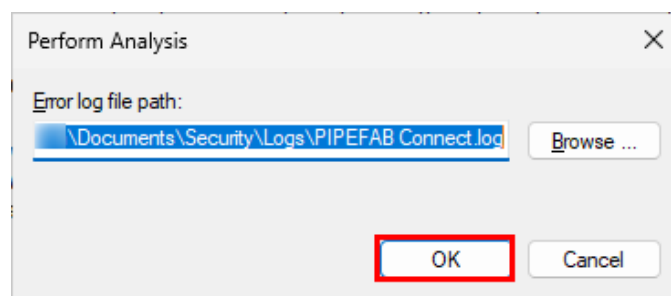


Abbildung 44: Log-Datei auswählen

- Nach der Analyse des Computer's kann unter System Services der *PBS PIPEFAB-Connect Import Worker Service (x64)* ausgewählt werden. Siehe die folgende Abbildung.

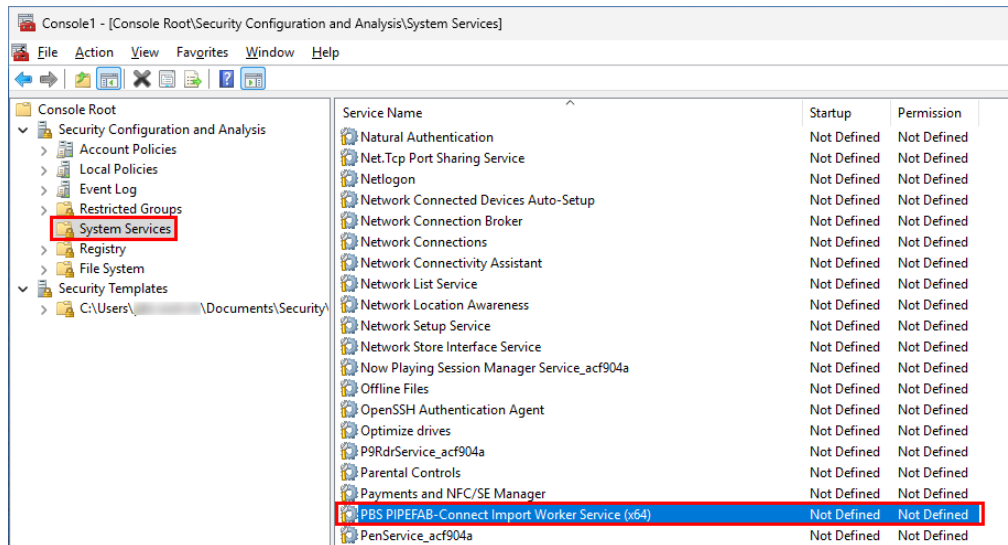


Abbildung 45: Auswahl des PBS PIPEFAB Connect Import Worker Service

- Mit einem Doppelklick auf den ausgewählten *PBS PIPEFAB-Connect Import Worker Service (x64)* werden die zugehörigen Security Einstellungen angezeigt. Siehe die folgende Abbildung.

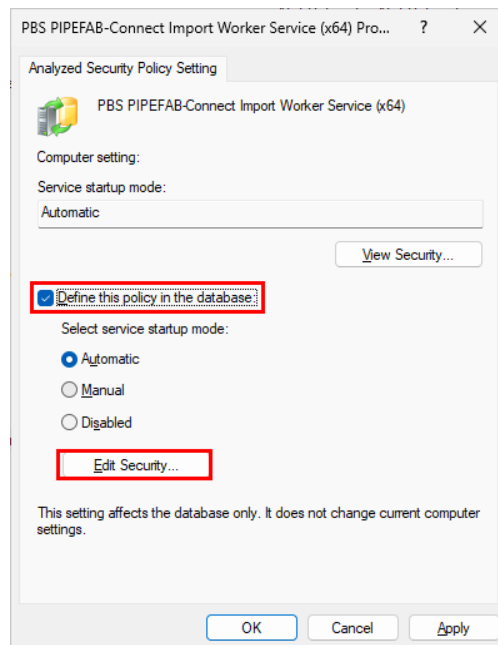


Abbildung 46: Sicherheitseinstellungen bearbeiten

- Zum Ändern der Sicherheitseinstellungen bitt den Hacken „Define this policy in the database“ setzen und den Button „Edit Security“ auswählen. Es wird eine Liste der Benutzer angezeigt. Zu dieser Liste muss der bei der Installation des PIPEFAB Connect Import Workers Service zugeordnete Benutzer hinzugefügt werden. Wählen Sie den Add-Button. Siehe folgende Abbildung.

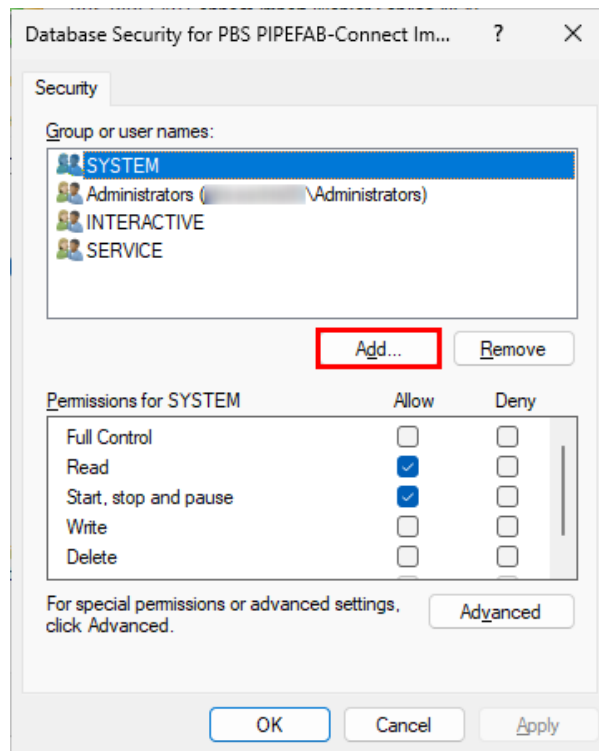


Abbildung 47: Benutzer hinzufügen

- Location auswählen und mit „Advanced“ fortfahren

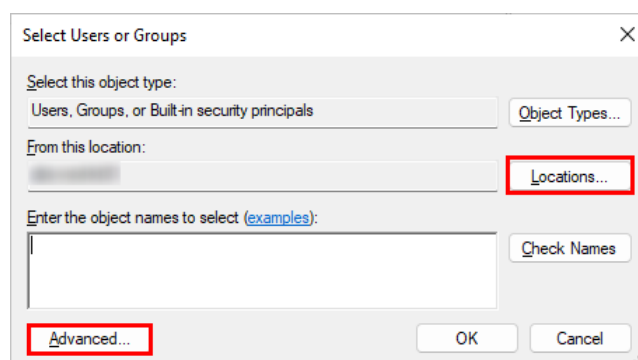


Abbildung 48: Location auswählen

- Benutzer suchen mit Hilfe des Buttons „Find Now“

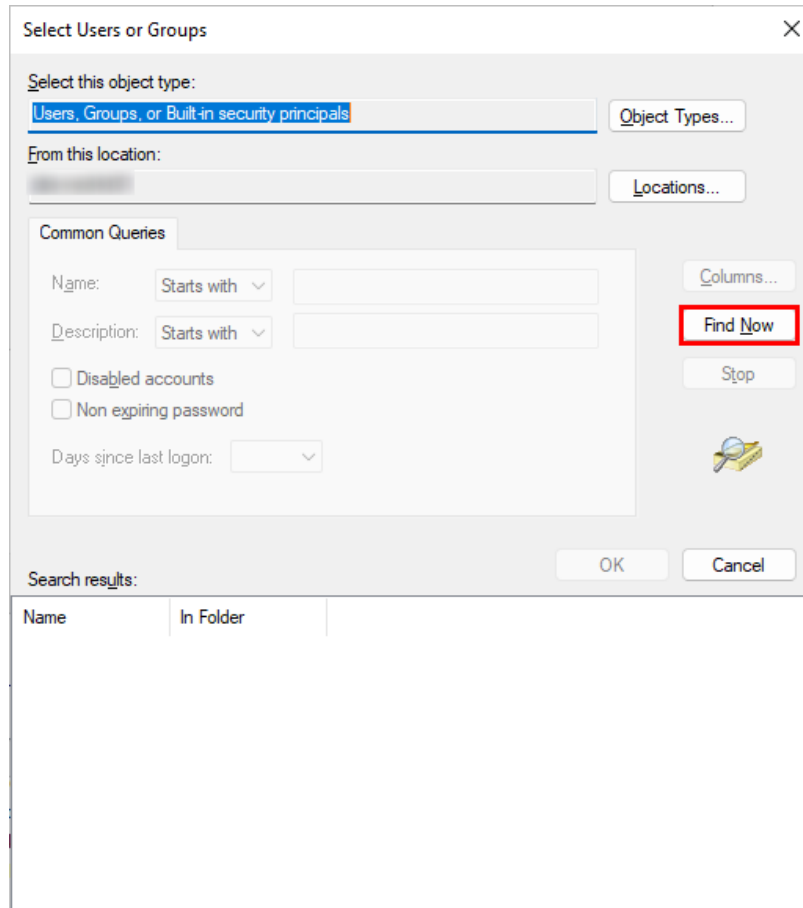


Abbildung 49: Benutzer suchen

- Aus der Liste der gefundenen Benutzer den Benutzer auswählen, der dem PIPEFAB Connect Service als Log-On Benutzer zugewiesen wurde. In der folgenden Darstellung ist das der Benutzer PBSConnectUser.

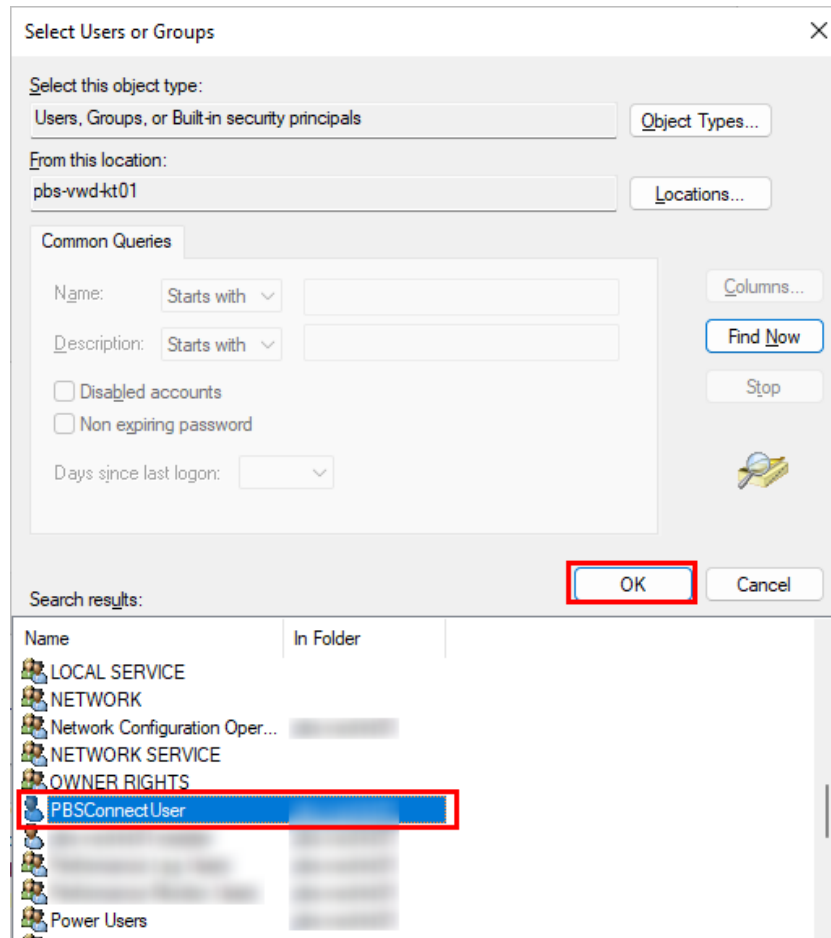


Abbildung 50: Benutzer auswählen

- Mit OK bestätigen.

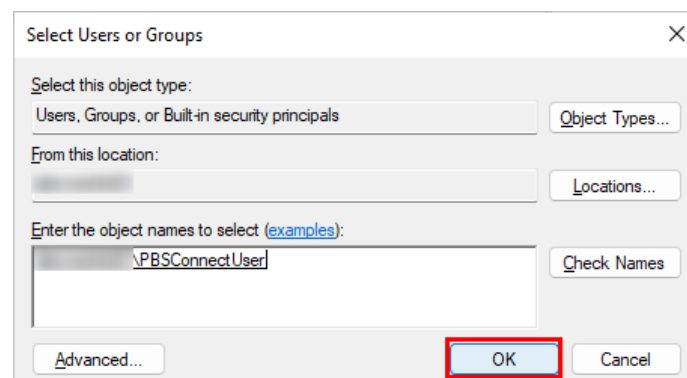


Abbildung 51: Ausgewählter Benutzer

- Berechtigungen für Read und „start, stop and pause“ erlauben.
- Mit OK bestätigen.

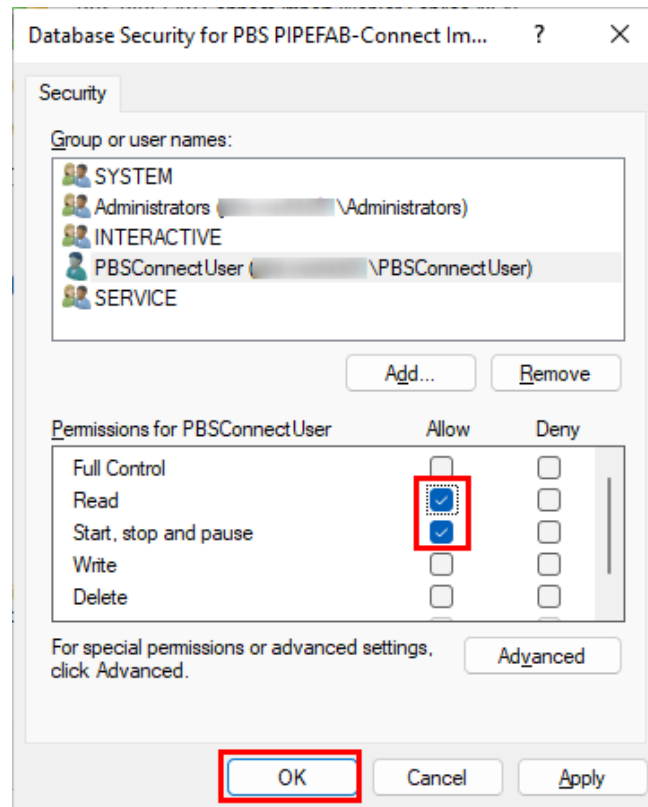


Abbildung 52: Berechtigung hinzufügen

- Die Service-Berechtigungen sind zu diesem Zeitpunkt nur in der Datenbank der MMC-Console gespeichert. Zu erkennen ist dieses am roten X vor dem Servicennamen. Siehe folgenden Screenshot.

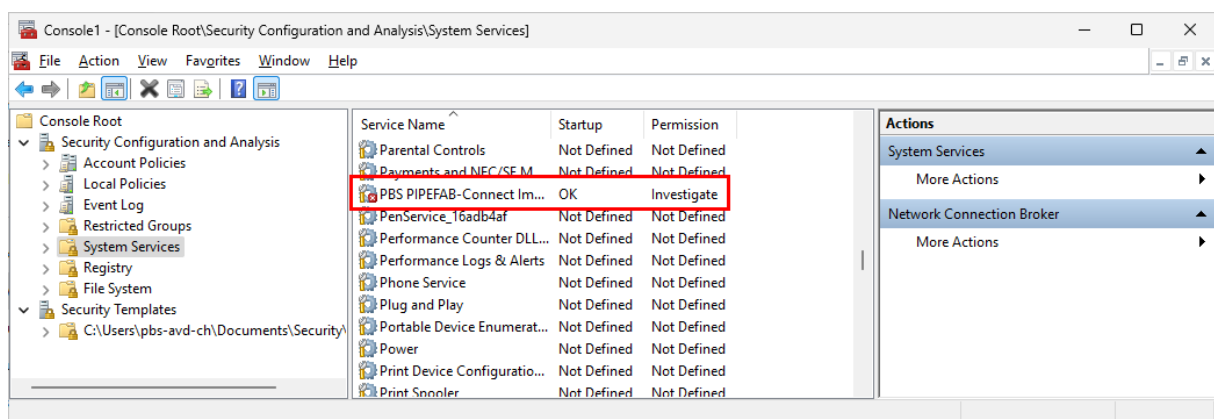


Abbildung 53: Änderungen wurden noch nicht in die Computerberechtigungen übertragen

- Das Übertragen der vorgenommenen Änderungen an den Computer erfolgt durch den Context-Menü-Eintrag „Configure Computer Now“. Siehe folgenden Screenshot.

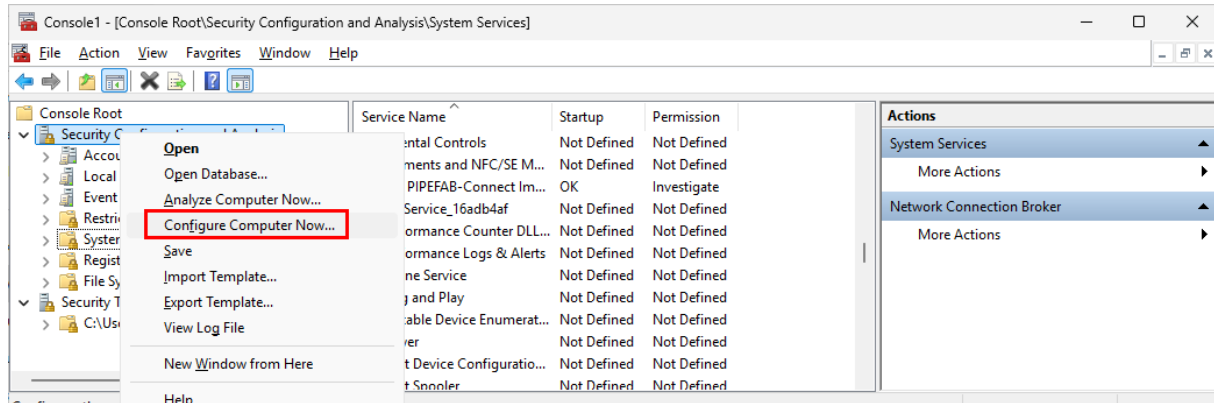


Abbildung 54: Konfiguration an den Computer übertragen

- Nach dem Übertragen der Änderungen erscheint vor dem Servicenamen ein grüner Haken. Damit sind die Berechtigungen des Services fertig konfiguriert und der Service kann gestartet werden.

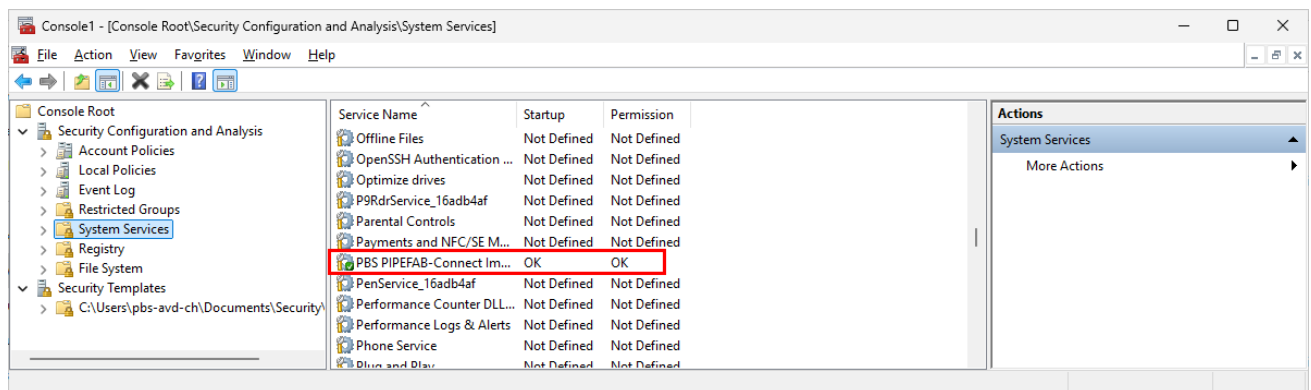


Abbildung 55: Konfiguration erfolgreich an den Computer übertragen

2. Möglichkeit: Mit Hilfe des Tools „ServiceSecurityEditor“ der Firma Core Technologies Consulting (<https://www.coretechnologies.com>).

Nach dem Start des ServiceSecurityEditors, muss als erstes der PBS PIPEFAB-Connect Import Worker Service (x64) in der Liste der Services (rot umrandet) ausgewählt werden.

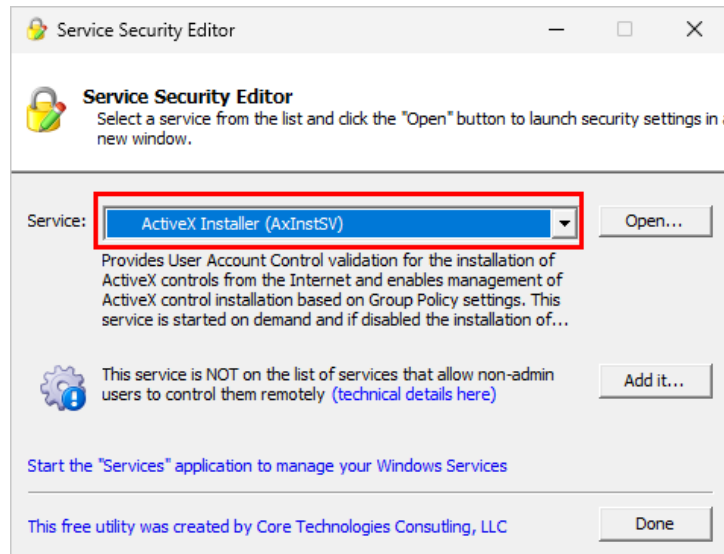


Abbildung 56: Service auswählen

Nach dem der Service ausgewählt wurde, kann im nächsten Schritt die Bearbeitung der Service-Berechtigungen erfolgen. Dazu den Button *Open* auswählen.

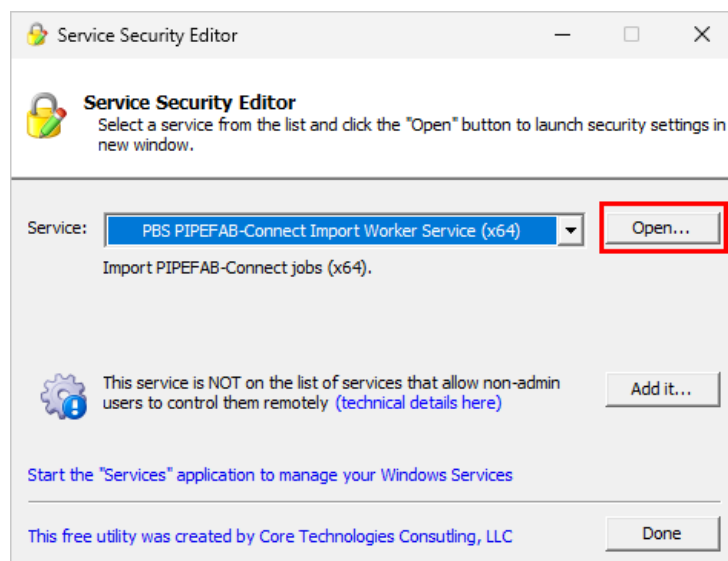


Abbildung 57: Service-Berechtigungen editieren

Als nächstes muss der User unter dessen Berechtigungen der PIPEFAB Connect ImportWorker Services ausgeführt wird, dem Service hinzugefügt werden. Dazu bitte den Button *Add* auswählen.

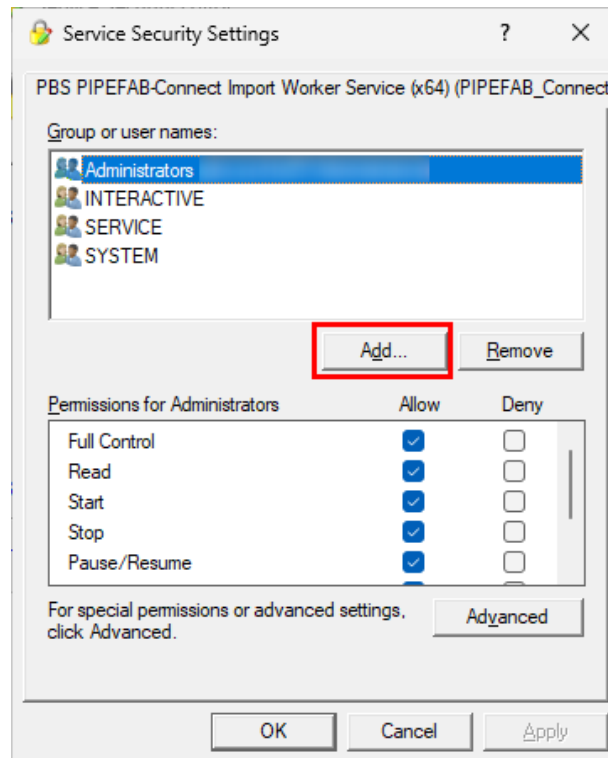


Abbildung 58: User hinzufügen

Den entsprechenden User (in diesem Beispiel lautet der User: PBSConnectUser) auf dem lokalen System suchen. Dazu bitte den Button *Advanced...* auswählen.

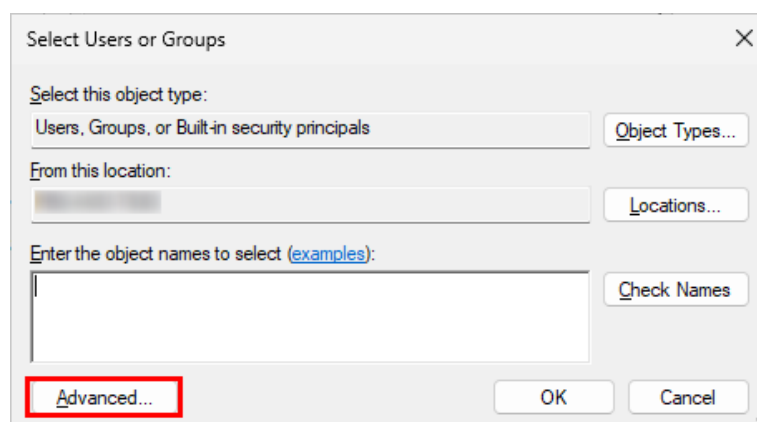


Abbildung 59: Suchverzeichnis auswählen

Dann den Button *Find Now* auswählen.

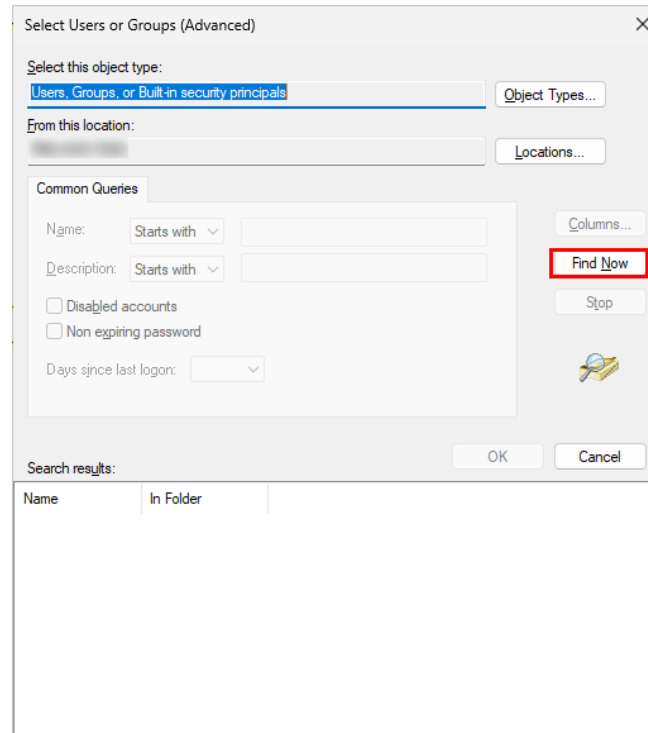


Abbildung 60: User suchen

User auswählen und mit OK bestätigen.

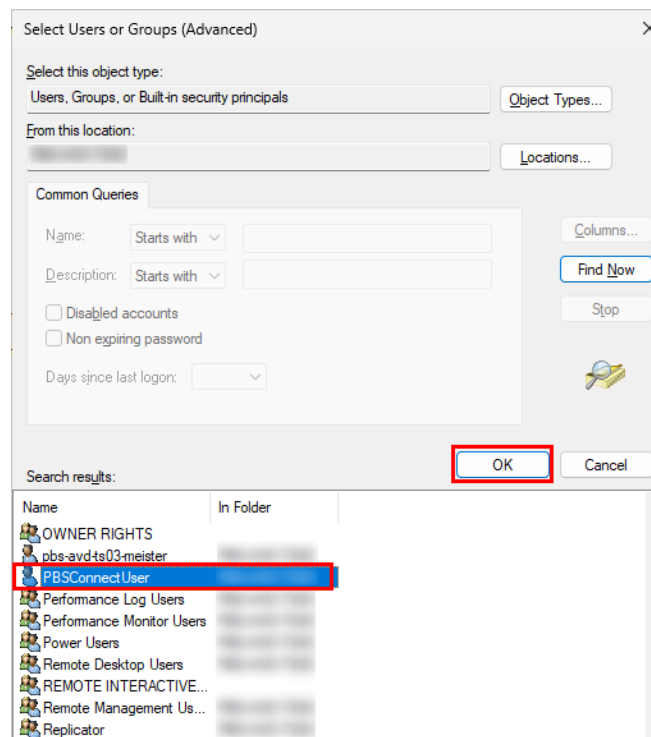


Abbildung 61: User auswählen

User mit OK dem Service hinzufügen.

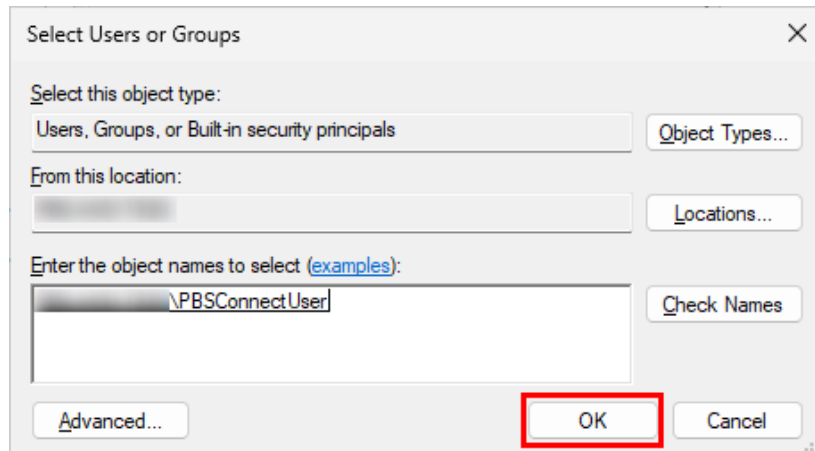


Abbildung 62: Ausgewählter User

Im nächsten Schritt werden die notwendigen Berechtigungen des Users gesetzt. Benötigt werden die Rechte zum Lesen, Starten und Stoppen des Services. Eingabe bitte mit OK bestätigen.

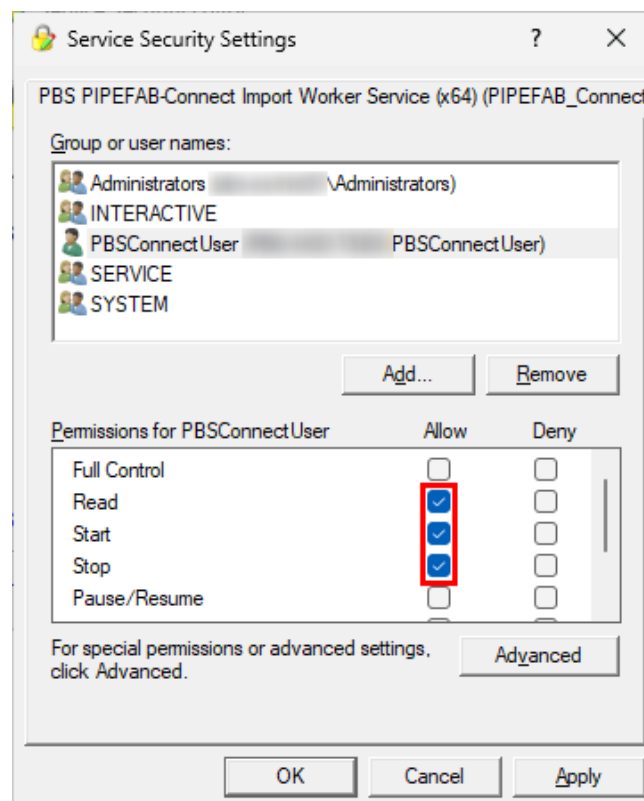


Abbildung 63: Notwendige Berechtigungen

Zum Abschluss werden die Einstellungen mit dem Button Done übernommen.

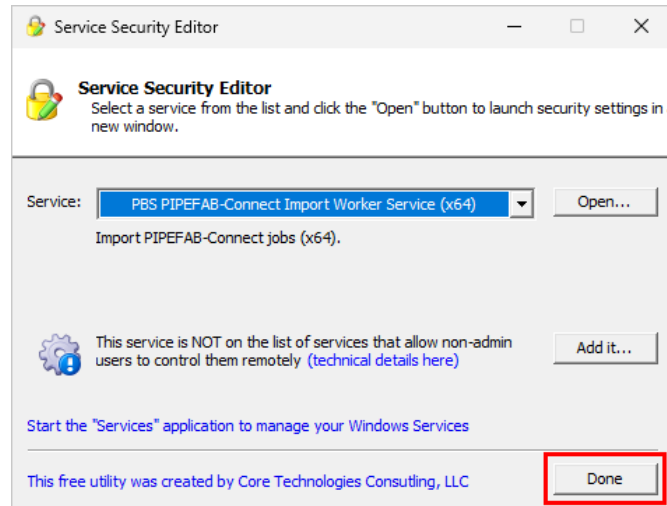


Abbildung 64: Einstellungen speichern

Nach dem Festlegen der Einstellungen kann der PIPEFAB Connect ImportWorker Service gestartet werden. Der Status des Services muss wie in der folgenden Abbildung dann auf Running stehen.

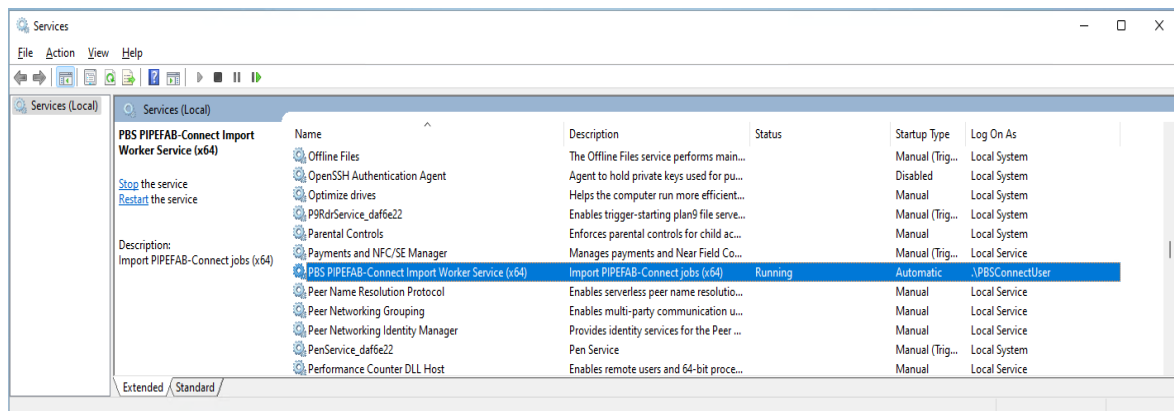


Abbildung 65: PIPEFAB Connect ImportWorker Service gestartet

5.2.10 Funktionstest

Der PIPEFAB Connect ImportWorker Service erzeugt im Verzeichnis C:\Temp ein Log-File. Eventuelle Konfigurationsfehler wie z.B. eine fehlerhafte Datenbankverbindung werden im Log-File abgelegt. Sollten keine Fehler mehr im Log-File vermerkt sein, kann über das PIPEFAB Connect WebApi ein Import-Auftrag initiiert werden.

5.2.11 PIPEFAB Connect ImportWorker-Service clean up

PIPEFAB Connect stellt mit dem ImportWorker-Service einen Windowsdienst zur Verfügung der den Import von CAD-Daten ermöglicht. Dabei verwendet der PIPEFAB Connect ImportWorker-Service wie auch der PIPEFAB Explorer lokale Caches zur Leistungssteigerung. Bei einer hohen Anzahl von Importen kann es deshalb abhängig von bestimmten Einflussfaktoren zu einem erhöhten Speicherverbrauch kommen. Um dem erhöhten Speicherverbrauch entgegen zu wirken, gibt es zwei Möglichkeiten:

- 1.) Automatische Speicherbereinigung beim Erreichen eines einstellbaren Speicherverbrauchs (CleanUpThreshold).
- 2.) Automatischer Neustart des PIPEFAB Connect ImportWorker-Services durch den Task-Scheduler.

CleanUpThreshold

Mit Hilfe des Parameters *CleanUpThresHold* kann die maximale Speicherverwendung des PIPEFAB Connect ImportWorker Services in Megabytes (MB) festgelegt werden. Bei Überschreitung des festgelegten Wertes, führt der PIPEFAB Connect ImportWorker Services selbstständig eine Bereinigung des verwendeten Speichers durch.

Der Parameter *CleanUpThresHold* kann in der Datei

TT.PipeFab.Connect.ImportWorker. Service.exe.config im Programmverzeichnis des PIPEFAB Connect ImportWorker Services konfiguriert werden. Der Standardwert für diesen Parameterwert beträgt 500 MB.

```
<appSettings>
  <add key="WorkerNumber" value="3"/>
  <add key="WorkerPool" value="TestPool"/>
  <add key="WorkerCycleTime" value="10000"/>
  <add key="BatchSize" value="5"/>
  <add key="CleanUpThreshold" value="500"/>
</appSettings>
```

Abbildung 66: Ausschnitt aus der Config-Datei

Task-Scheduler

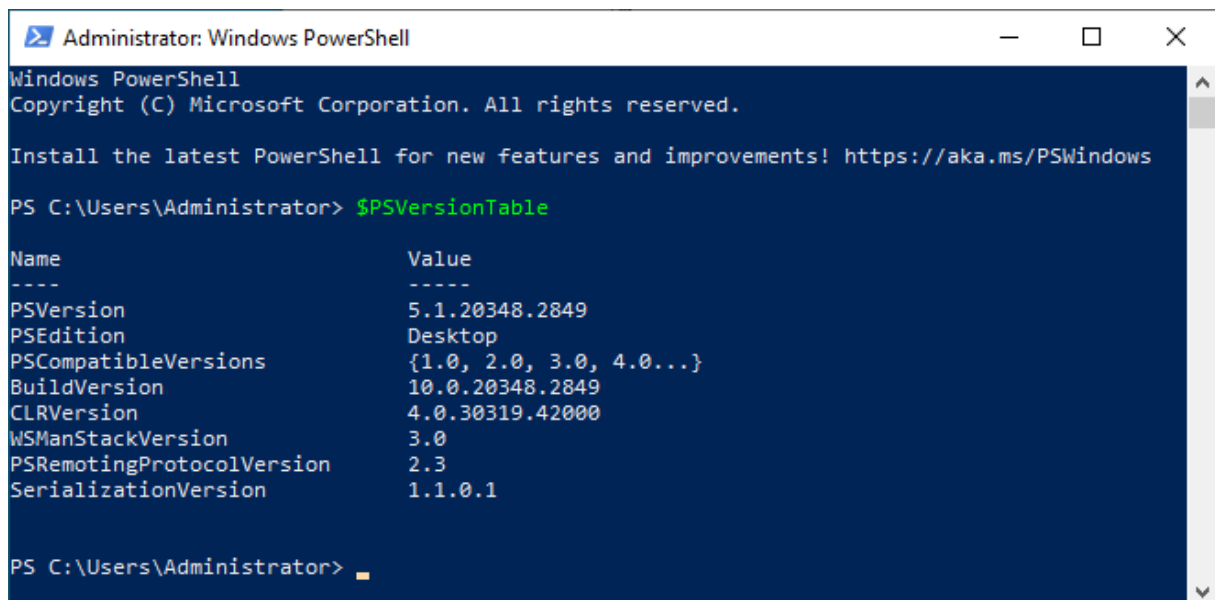
Der ImportWorker-Service Windowsdienst kann z.B. alle 24 Stunden unabhängig von derzeitigen Speicherverbrauch automatisiert neu gestartet werden. Dadurch werden nicht benötigte Ressourcen freigegeben.

Im Folgenden ist die Einrichtung und Ausführung eines Powershell-Scripts für den Windows Task Scheduler (Aufgabenplaner) beschrieben. Die Vorgehensweise wurde auf Basis der englischen Version des Microsoft Servers 2022 erstellt. Unterschiede zu anderen Windows Server Versionen und Sprachen sind möglich.

Windows Powershell Version prüfen und ggf. installieren

Empfohlen wird die Installation von Windows Powershell in der Version 7.x.

Zum Prüfen der installierten Powershell Version kann in einem geöffneten Powershell-Eingabefenster der Befehl „\$PSVersionTable“ ausgeführt werden. Die folgende Abbildung zeigt das Ergebnis der Versionsprüfung.



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Administrator> $PSVersionTable

Name                           Value
----                           -
PSVersion                      5.1.20348.2849
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.20348.2849
CLRVersion                     4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1

PS C:\Users\Administrator>
  
```

\$\$

Abbildung 67: Powershell Versionsprüfung

In dem vorherigen dargestellten Fall, empfiehlt sich ein Update auf die Powershell Version 7.x.

Microsoft stellt dafür ein Installationsprogramm zur Verfügung.

Anlegen der Task im Task Scheduler (Aufgabenplaner)

Den Task Scheduler durch Eingabe des Namens in der Suchleiste suchen und starten wie in der folgenden Abbildung dargestellt.

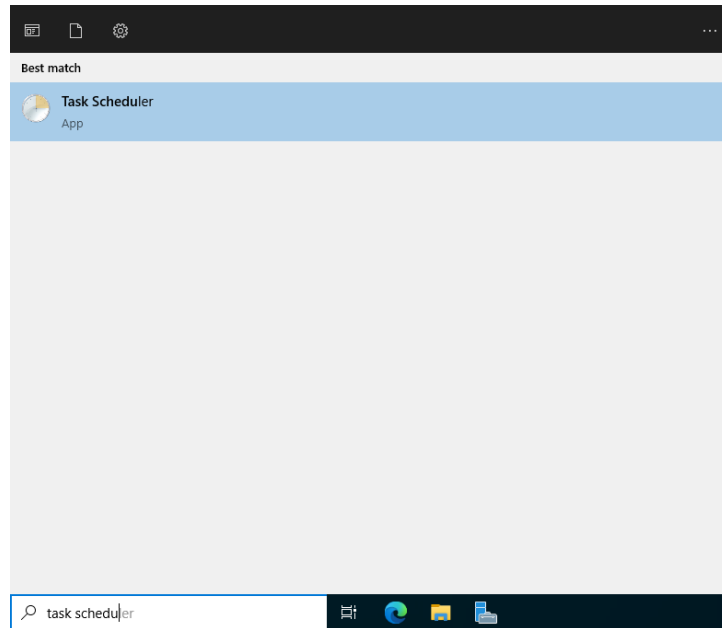


Abbildung 68: Suche des Task Scheduler

Wenn der Task Scheduler gestartet ist, sollte zu besseren Übersicht als erstes ein neues Verzeichnis in der links dargestellten Baumstruktur angelegt werden. Dazu muss mit der rechten Maustaste das Context-Menü geöffnet werden und der Punkt „New Folder“ ausgewählt werden.

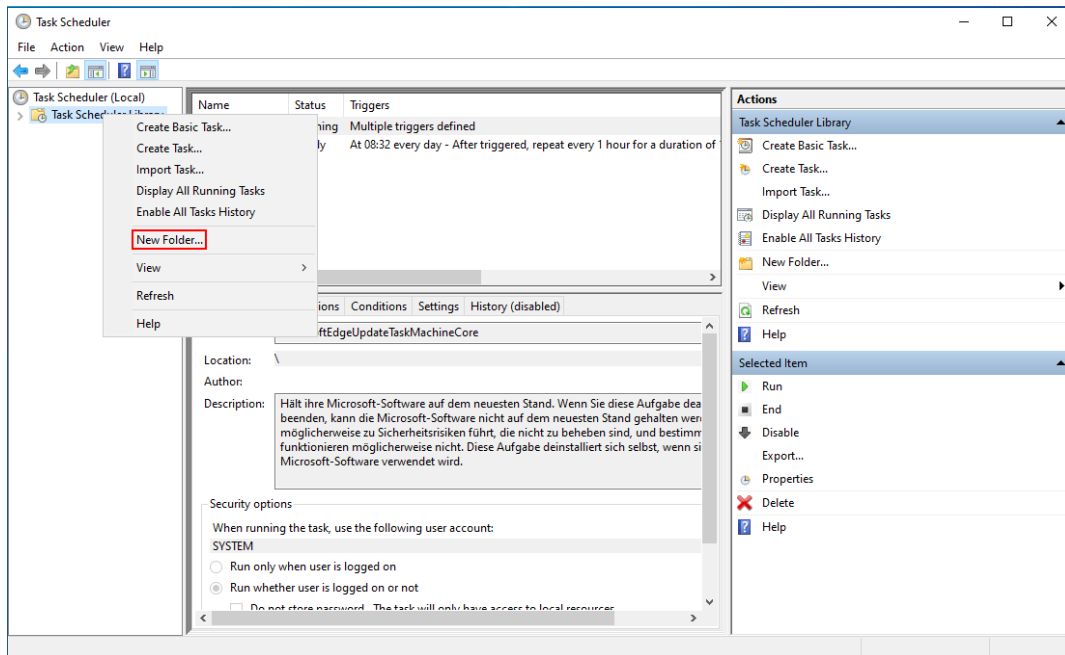


Abbildung 69: Neues Verzeichnis im Task Scheduler anlegen

Als Name des neuen Verzeichnisses kann z.B. „PBS PIPEFAB Connect“ gewählt werden.

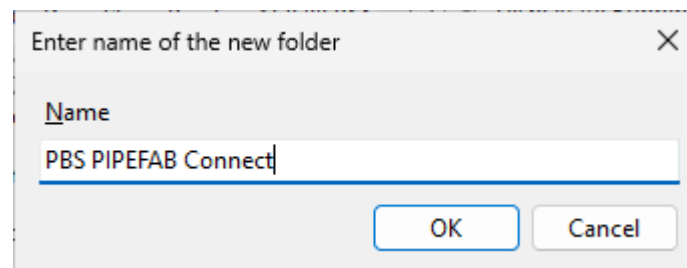


Abbildung 70: Eingabe des Verzeichnisnamens

Als nächstes muss ein neuer Task (Aufgabe) angelegt werden. Dazu wird das Context-Menü des neu angelegten Verzeichnisses „PBS PIPEFAB Connect“ geöffnet und der Menüpunkt „Create Task“ ausgewählt werden.

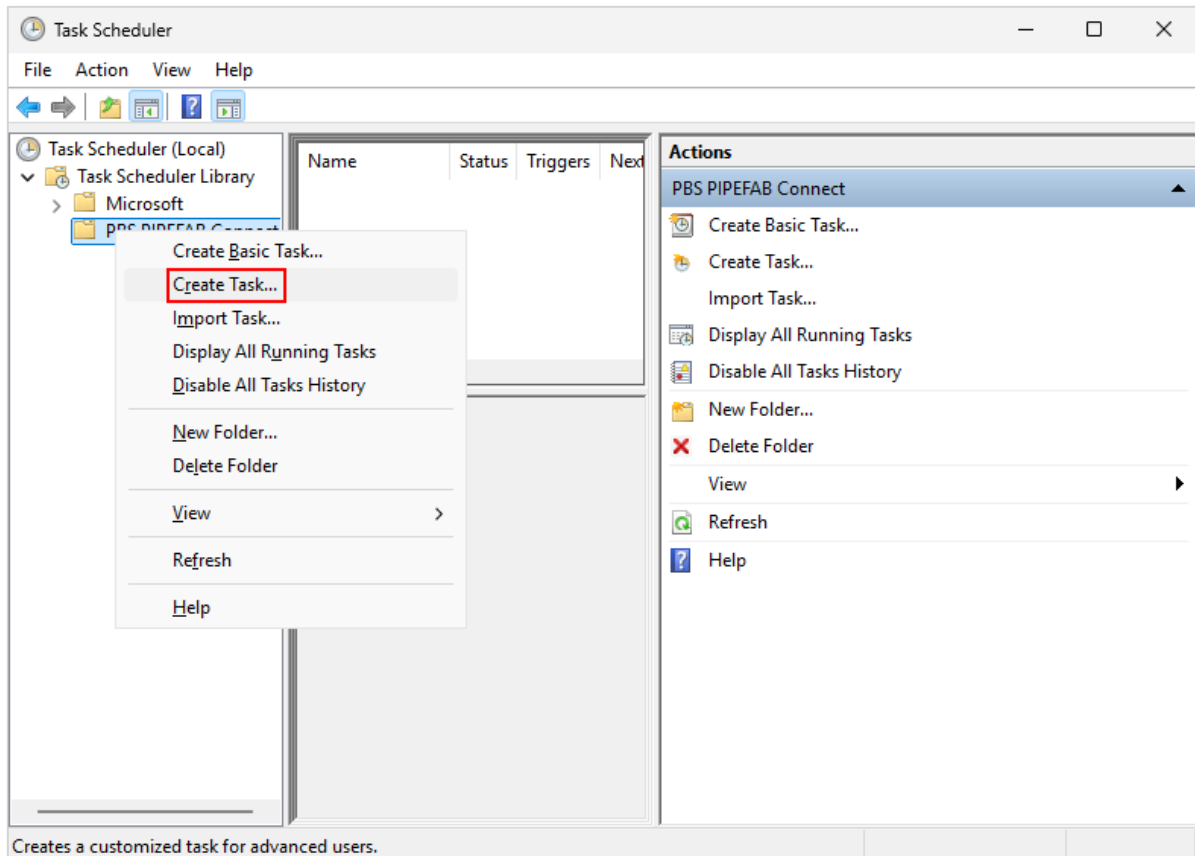


Abbildung 71: Neue Task (Aufgabe) erstellen

Im nächsten Dialog wird der Name für die neue Aufgabe angegeben. In diesem Fall lautet der Name „PIPEFAB Connect ImportWorker restart“. Zudem muss ein Benutzer ausgewählt werden, der das Recht besitzt Windowsdienste zu beenden und zu starten. Der Benutzer kann mit Hilfe des Buttons „Change User or Group“ ausgewählt werden.

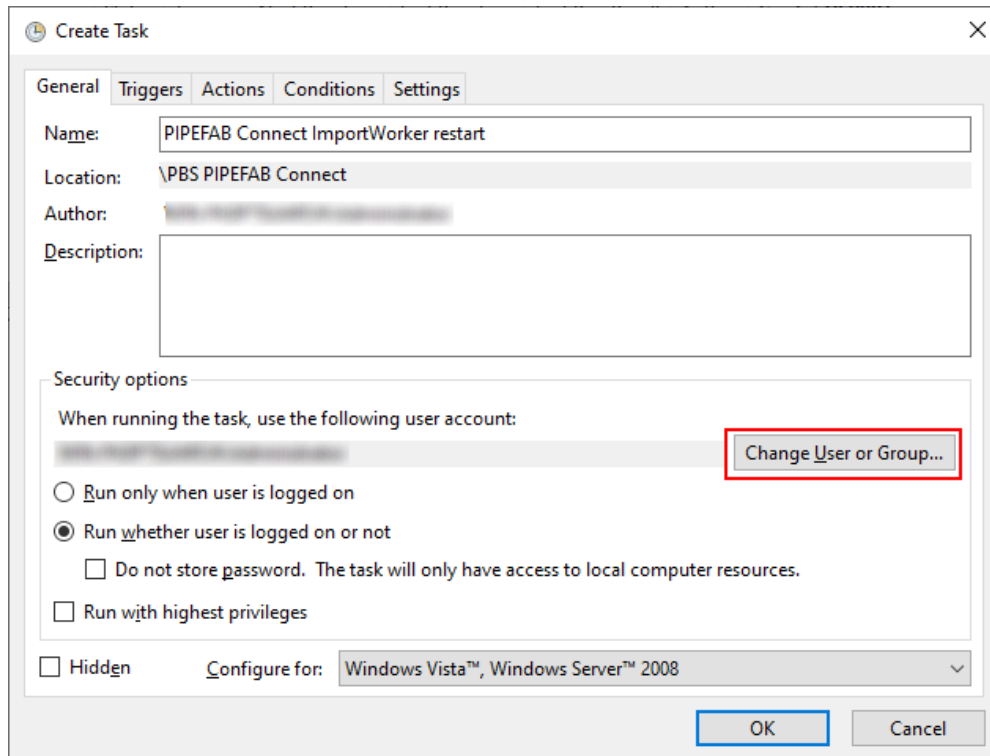


Abbildung 72: Neue Aufgabe anlegen

Auf dem Tabreiter „Triggers“ muss ein neuer Trigger angelegt werden. Dazu den Button „New“ auswählen.

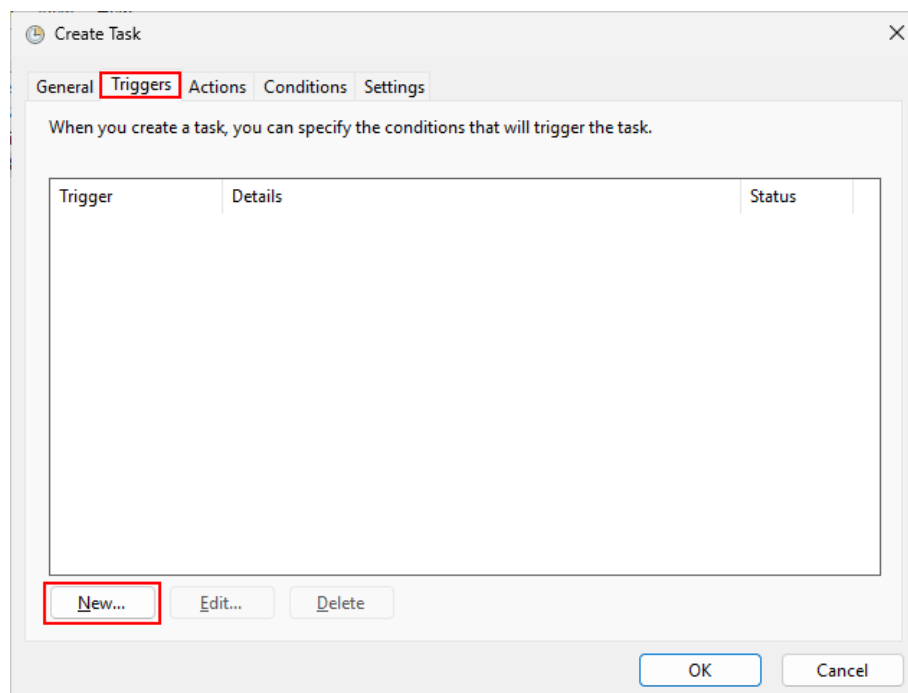
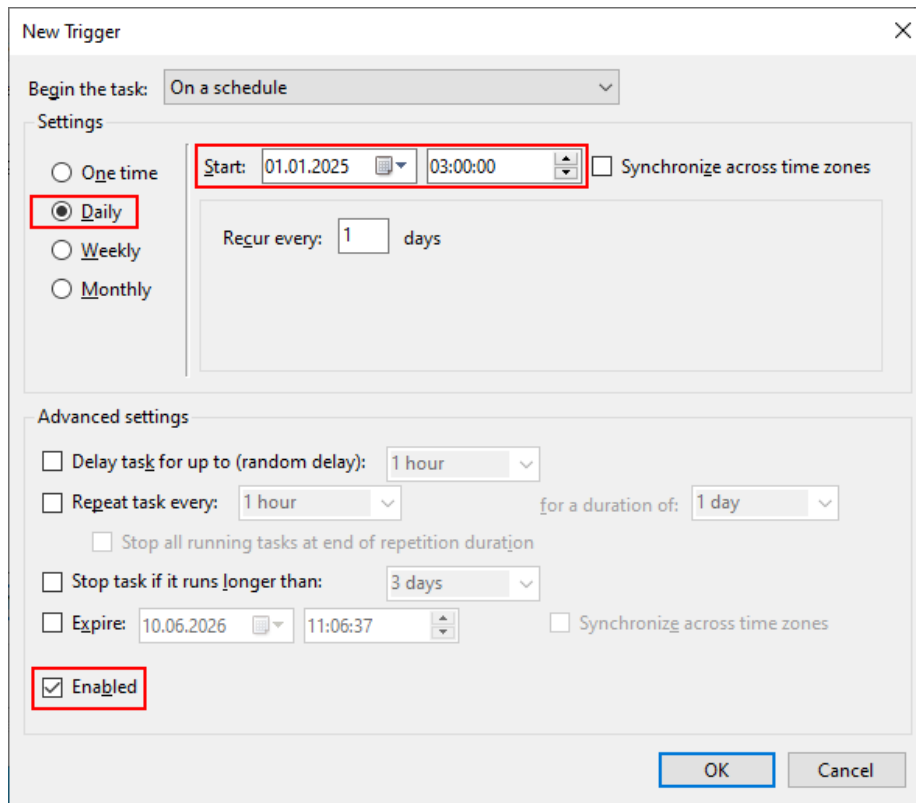


Abbildung 73: Neuen Trigger hinzufügen

Als Trigger ist der Zeitpunkt anzugeben an dem der PIPEFAB Connect ImportWorker-Service beendet und neu gestartet werden soll.

In diesem Beispiel wird der PIPEFAB Connect ImportWorker-Service täglich um 3 Uhr in der Nacht beendet und neu gestartet. Wichtig ist hierbei, dass der Trigger auch enabled (scharfgeschaltet) ist. Die folgende Abbildung zeigt die für dieses Beispiel notwendigen Einstellungen.



New Trigger

Begin the task: On a schedule

Settings

☐ One time

☒ Daily

☐ Weekly

☐ Monthly

Start: 01.01.2025 03:00:00

☐ Synchronize across time zones

Repeat every: 1 days

Advanced settings

☐ Delay task for up to (random delay): 1 hour

☐ Repeat task every: 1 hour for a duration of: 1 day

☐ Stop all running tasks at end of repetition duration

☐ Stop task if it runs longer than: 3 days

☐ Expire: 10.06.2026 11:06:37

☐ Synchronize across time zones

☒ Enabled

OK Cancel

Abbildung 74: Trigger um 3 Uhr

Nach dem Anlegen des neuen Triggers, wird als nächstes die zum Trigger passende Action erfasst. Dazu bitte auf den Tabreiter „Actions“ wechseln und den Button „New“ auswählen.

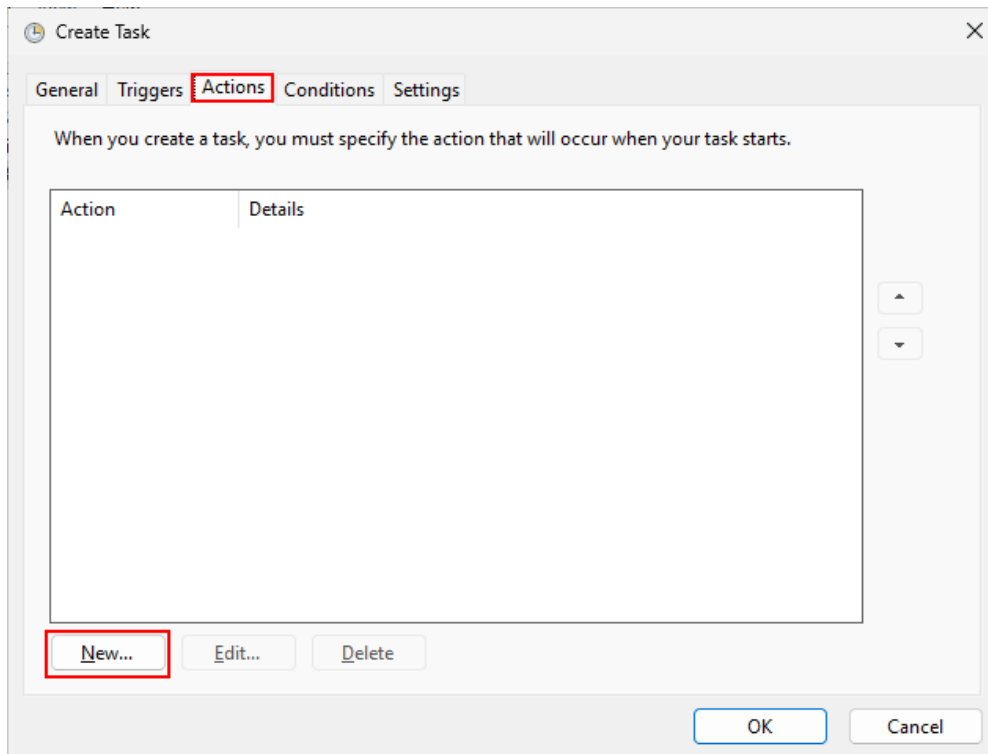


Abbildung 75: Neue Action anlegen

Als Action wird die Ausführung des folgenden Powershell-Scriptes angelegt.

```
net stop PIPEFAB_Connect_ImportWorkerService
timeout /T 20
net start PIPEFAB_Connect_ImportWorkerService
```

Abbildung 76: Powershell-Script

Das Powershell-Script beendet den PIPEFAB Connect ImportWorker-Service und startet diesen mit einer Verzögerung von 20 Sekunden neu.

Die drei oben abgebildeten Textzeilen des Powershell-Scriptes werden dazu in einer Datei mit dem Namen „PBS_PIPEFAB_Connect_Import_Worker_restart.ps1“ im Verzeichnis C:\Temp gespeichert.

In dem Dialog zur Anlage der neuen Action wird als Aktion „Start a program“ ausgewählt.

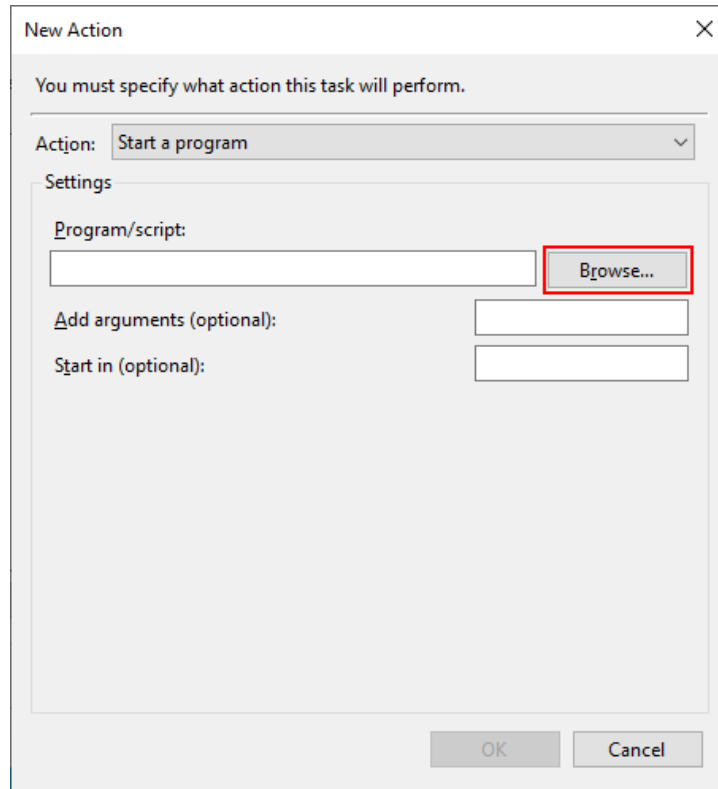


Abbildung 77: Neue Action anlegen

Über den „Browse“-Button wird die Datei zum Starten der Windows-Powershell ausgewählt. Das ist bei der Verwendung von Powershell 7 das Programm „pwsh.exe“ im Verzeichnis „C:\Program Files\Powershell\7\“. Das Verzeichnis kann auf anderen Systemen variieren.

Die folgende Abbildung zeigt die Auswahl des zu startenden Programms. Als Start-Argument wird der Powershell ein Kommando zum Ausführen des weiter oben beschriebenen Scripts mitgegeben.

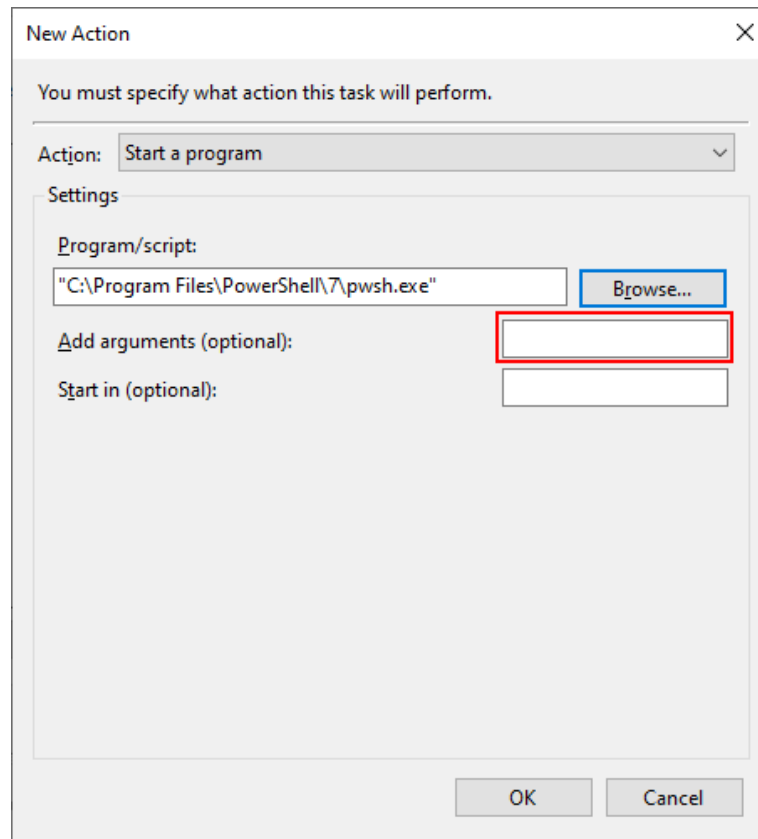


Abbildung 78: Neue Action, Powershell starten

Dazu ist bei „Add arguments“ der folgende Text einzutragen:

`-command "C:\temp\PBS_PIPEFAB_Connect_Import_Worker_restart.ps1"`

Dieser sorgt dafür, dass das Powershell-Script zum Beenden und Starten des PIPEFAB Connect ImportWorker-Services aus dem Verzeichnis „C:\Temp“ ausgeführt wird. Sollte das Script in einem anderen Verzeichnis liegen, ist das Kommando entsprechend anzupassen.

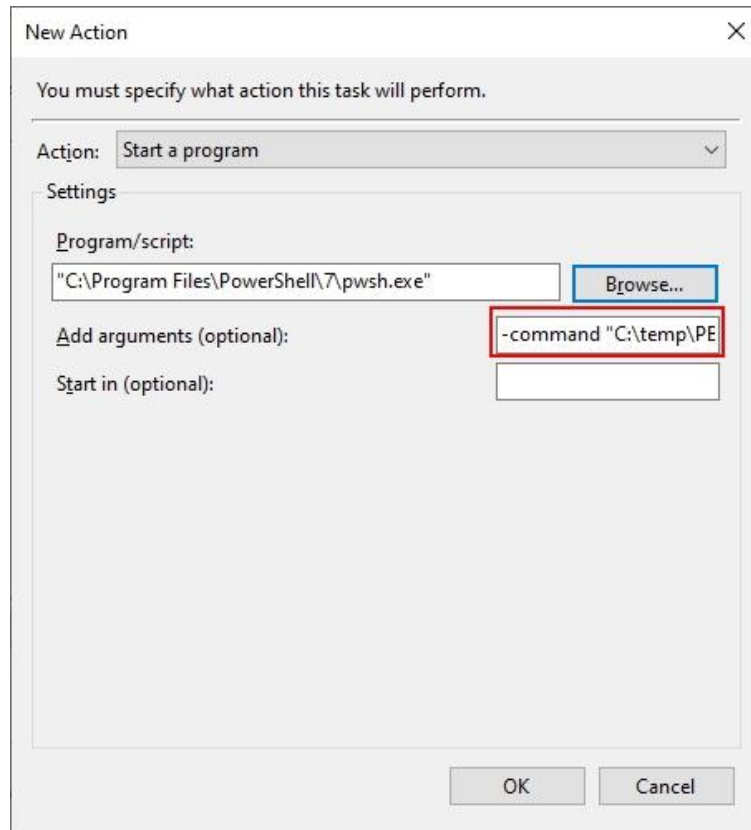


Abbildung 79: Neue Action, mit Command Arguments

Nach der Eingabe der Argumente kann die Anlage der neuen Aktion mit dem OK-Button abgeschlossen werden.

Auf dem Tabreiter „Conditions“ sind soweit erforderlich die folgenden Einstellungen vorzunehmen.

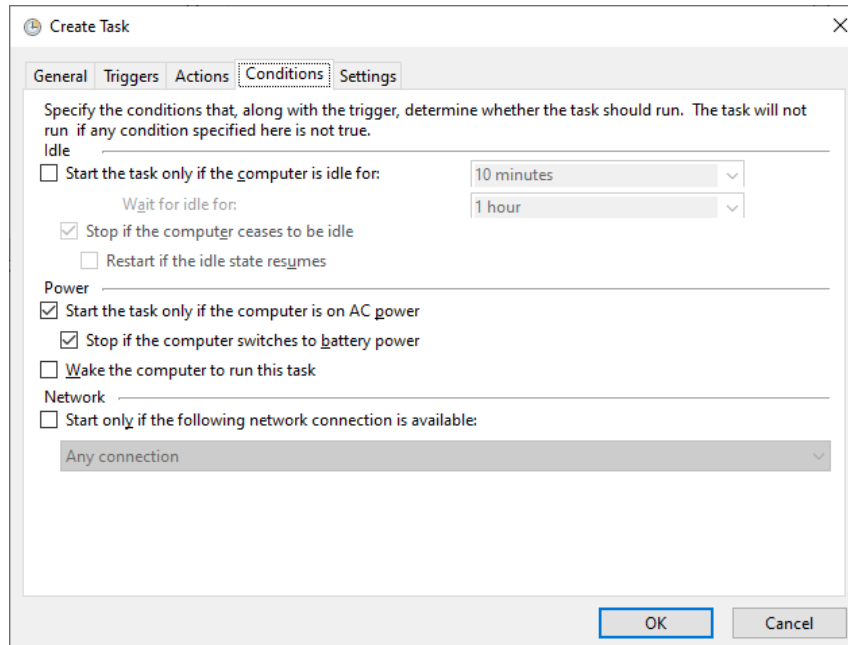


Abbildung 80: Tabreiter Conditions

Auf dem Tabreiter „Settings“ können die folgenden Einstellungen vorgenommen werden.

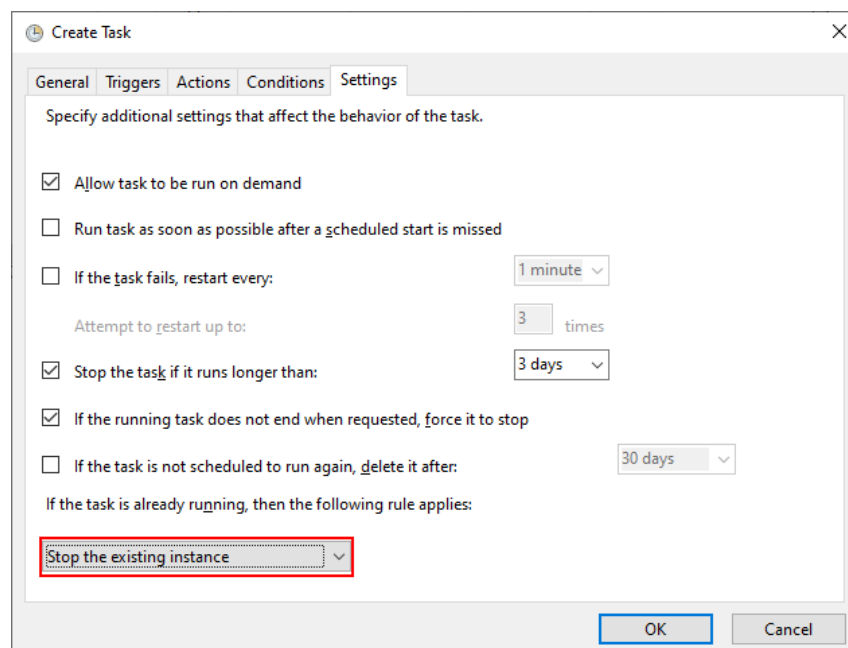


Abbildung 81: ii

Mit Auswahl des OK-Buttons wird die neue Aufgabe angelegt.

Task History

Wenn gewünscht kann das Log (Task History) für den Task Scheduler aktiviert werden. Dazu den Punkt „Enable All Task History“ auswählen.

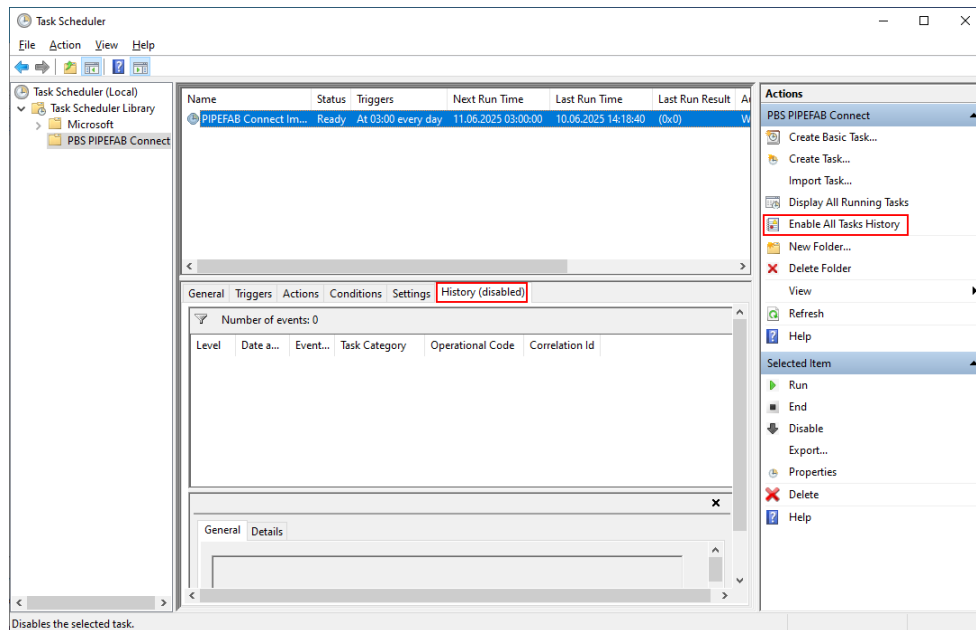


Abbildung 82: Log einschalten (Enable All Task History)

Bei eingeschaltetem Log kann der Status der Aufgabenausführung überprüft werden.

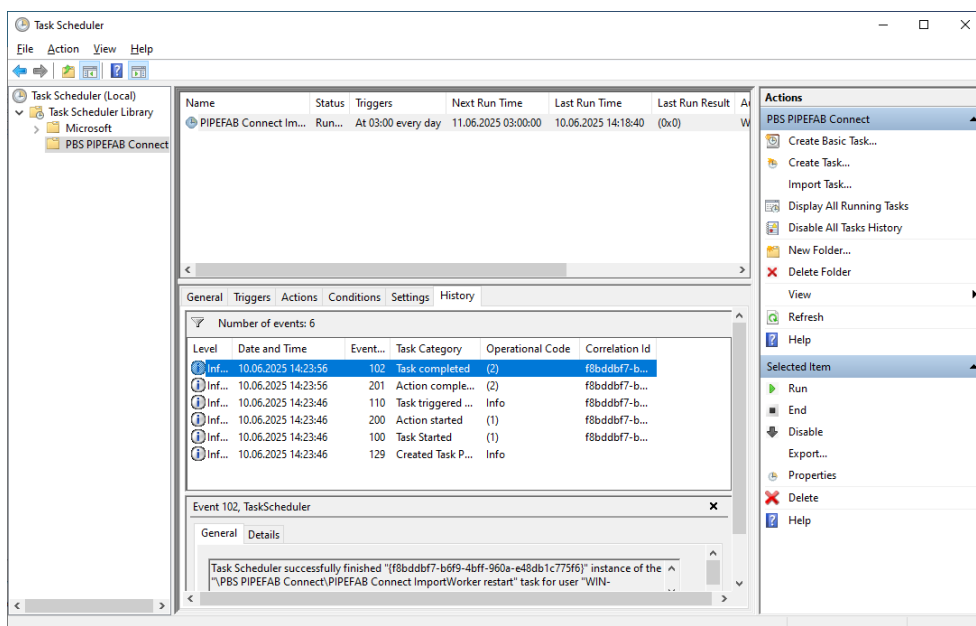


Abbildung 83: Aktiviertes Log (History)

5.2.12 Zugriff auf Netzwerklauferwerke (UNC-Pfad)

Für den Zugriff auf Netzwerklauferwerke gelten für den PIPEFAB Connect ImportWorker Service, der als Windows-Dienst läuft, die vom Windows-System festgelegten Regeln.

- a) Ein Zugriff auf vorhandene Netzwerklauferwerke muss mittels UNC-Pfadangabe erfolgen. Die Angabe von Netzwerklauferwerke über den zugeordneten Laufwerksbuchstaben ist nicht möglich.

Das Ermitteln eines UNC-Pfades für ein vorhandenes Netzlaufwerk kann mit Hilfe des Kommandos „net use“ erfolgen.

Beispiel: Ausführung des Kommandos „net use“ in einer Eingabeaufforderung auf einem System bei dem der Laufwerksbuchstabe „S:“ auf ein Netzlaufwerk gemapped ist.

```
Microsoft Windows [Version 10.0.22621.5189]
(c) Microsoft Corporation. All rights reserved.

C:\Users\...>net use
New connections will be remembered.
```

Status	Local	Remote	Network
OK	S:	\\storage\ue.file.core.windows.net\storage	Microsoft Windows Network

```
The command completed successfully.
```

Abbildung 84: Ausführung von net use

In diesem Beispiel ergibt sich für das Netzwerklauferwerk „S:“ der UNC-Pfad:

<\\storage\ue.file.core.windows.net\storage>.

Dieser Pfad muss in den CadCon-Import-Einstellung anstatt der Angabe des Netzwerklauferwerks S: angegeben werden.

In der folgenden Abbildung sind die Pfade für die Ausgabe, die importierten und fehlerhaften PCF-Dateien mittels UNC-Pfadangaben angegeben.

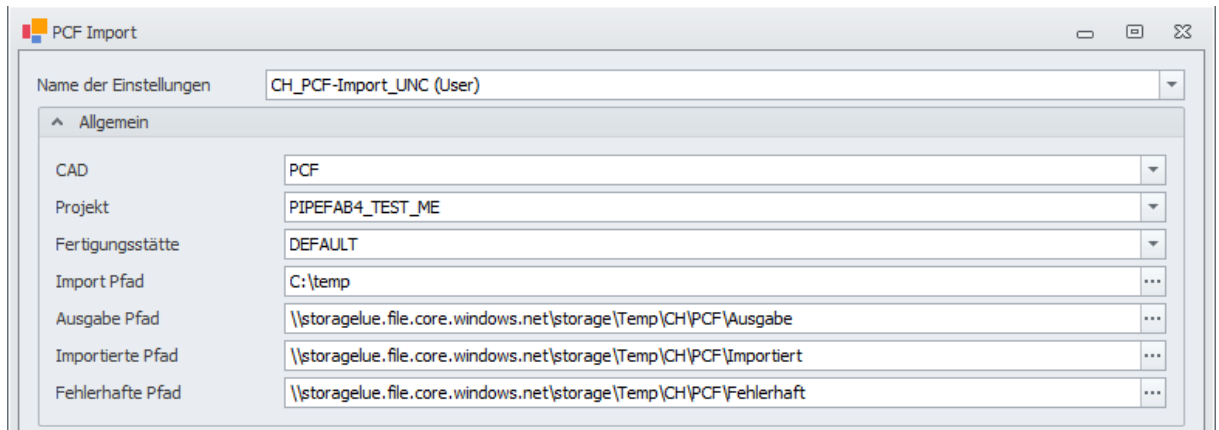


Abbildung 85: UNC-CadCon-Einstellungen

- b) Der Windows-Benutzer in dessen Kontext der PIPEFAB ImportWorker-Service (Windows-Dienst) läuft, muss über die Zugriffsberechtigungen auf das Netzwerklaufwerk (UNC-Pfad) verfügen.

6 Beispielkonfigurationen PIPEFAB Connect ImportWorker

6.1 Ein ImportWorker

Szenario:

Es ist nur ein PIPEFAB Connect ImportWorker Service installiert. Der Worker erhält die Nummer 1 (WorkerNumber). Der Worker soll alle 10 Sekunden (WorkerCycleTime) nach neuen Aufträgen in der Datenbank nachschauen. Der Worker soll beim Vorliegen neuer Aufträge max. 5 Aufträge (BatchSize) am Stück aus der Datenbank lesen und dann nacheinander abarbeiten. Sofern mehr als fünf Aufträge in der Datenbank vorhanden sind, werden die zusätzlichen Aufträge ohne Verzögerung anschließend verarbeitet (gestaffelt in Gruppen zu je max. 5 Aufträgen).

Sind keine weiteren Aufträge vorhanden, wird der Worker alle 10 Sekunden nach neuen Aufträgen in der Datenbank nachschauen.

```
<appSettings>
  <add key="WorkerNumber" value="1"/>
  <add key="WorkerPool" value=""/>
  <add key="WorkerCycleTime" value="10000"/>
  <add key="BatchSize" value="5"/>
</appSettings>
```

Abbildung 86: Ein Worker Konfiguration

6.2 Mehrere ImportWorker

Szenario:

Es sind drei PIPEFAB Connect ImportWorker Service installiert. Die Worker sind mit den Nummern 1, 2 und 3 (WorkerNumber) bezeichnet. Die Worker sind keinem Worker-Pool zugeordnet (WorkerPool = „“) Die Worker sollen alle 5 Sekunden (WorkerCycleTime) nach neuen Aufträgen in der Datenbank suchen. Beim Vorliegen neuer Aufträge sollen die Worker nur einen Auftrag (BatchSize = 1) aus der Datenbank lesen und für die Verarbeitung reservieren. Nach der Verarbeitung prüfen die Worker ob neue Aufträge vorliegen.

Sind keine weiteren Aufträge vorhanden, wird jeder Worker wiederholend alle 5 Sekunden nach neuen Aufträgen in der Datenbank suchen.

Folgend ist der Inhalt der drei Konfigurationsdateien der PIPEFAB Connect ImportWorker Services abgebildet.

```
<appSettings>
  <add key="WorkerNumber" value="1"/>
  <add key="WorkerPool" value=""/>
  <add key="WorkerCycleTime" value="5000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 87: Konfiguration Worker 1

```
<appSettings>
  <add key="WorkerNumber" value="2"/>
  <add key="WorkerPool" value=""/>
  <add key="WorkerCycleTime" value="5000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 88: Konfiguration Worker 2

```
<appSettings>
  <add key="WorkerNumber" value="3"/>
  <add key="WorkerPool" value=""/>
  <add key="WorkerCycleTime" value="5000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 89: Konfiguration Worker 3

6.3 Zwei Worker-Pool's mit je zwei ImportWorkern

Szenario:

Es existieren zwei Worker-Pools mit dem Namen „Pool1“ und „Pool2“. Für jeden Worker-Pool sind je zwei PIPEFAB Connect ImportWorker Service's installiert. Die Worker sind fortlaufend von 1-4 nummeriert (WorkerNumber). Alle Worker sollen nach 7 Sekunden Wartezeit (WorkerCycleTime) erneut nach neuen Aufträgen in der Datenbank suchen. Dabei suchen die Worker nur die Aufträge die Ihrem Worker-Pool zugewiesen wurden. Die Worker sollen beim Vorliegen neuer Aufträge nur einen Auftrag (BatchSize = 1) aus der Datenbank lesen und für die Verarbeitung reservieren. Nach der Verarbeitung prüft jeder Worker ob neue Aufträge für ihn vorliegen.

Sind keine weiteren Aufträge vorhanden, werden alle Worker jeweils nach 7 Sekunden nach neuen Aufträgen in der Datenbank suchen.

Konfigurationen im WorkerPool „Pool1“:

```
<appSettings>
  <add key="WorkerNumber" value="1"/>
  <add key="WorkerPool" value="Pool1"/>
  <add key="WorkerCycleTime" value="7000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 90: Konfiguration Worker 1, WorkerPool: Pool1

```
<appSettings>
  <add key="WorkerNumber" value="2"/>
  <add key="WorkerPool" value="Pool1"/>
  <add key="WorkerCycleTime" value="7000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 91: Konfiguration Worker 2, WorkerPool: Pool1

Konfigurationen im WorkerPool „Pool2“:

```
<appSettings>
  <add key="WorkerNumber" value="3"/>
  <add key="WorkerPool" value="Pool2"/>
  <add key="WorkerCycleTime" value="7000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 92: Konfiguration Worker 3, WorkerPool: Pool2

```
<appSettings>  
  <add key="WorkerNumber" value="4"/>  
  <add key="WorkerPool" value="Pool2"/>  
  <add key="WorkerCycleTime" value="7000"/>  
  <add key="BatchSize" value="1"/>  
</appSettings>
```

Abbildung 93: Konfiguration Worker 4, WorkerPool: Pool2

6.4 Ein Worker-Pool mit drei ImportWorkern

Szenario:

Es sind drei PIPEFAB Connect ImportWorker Service installiert. Die Worker sind mit den Nummern 1, 2 und 3 (WorkerNumber) bezeichnet. Die Worker sind dem Worker-Pool „CadPool“ zugeordnet (WorkerPool = „CadPool“) Die Worker sollen alle 10 Sekunden (WorkerCycleTime) nach neuen Aufträgen in der Datenbank suchen. Beim Vorliegen neuer Aufträge sollen die Worker nur einen Auftrag (BatchSize = 1) aus der Datenbank lesen und für die Verarbeitung reservieren. Nach der Verarbeitung prüfen die Worker ob neue Aufträge vorliegen.

Sind keine weiteren Aufträge vorhanden, wird jeder Worker wiederholend alle 10 Sekunden nach neuen Aufträgen in der Datenbank suchen.

Folgend ist der Inhalt der drei Konfigurationsdateien der PIPEFAB Connect ImportWorker Services abgebildet.

```
<appSettings>
  <add key="WorkerNumber" value="1"/>
  <add key="WorkerPool" value="CadPool"/>
  <add key="WorkerCycleTime" value="10000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 94: Konfiguration Worker 1, WorkerPool: CadPool

```
<appSettings>
  <add key="WorkerNumber" value="2"/>
  <add key="WorkerPool" value="CadPool"/>
  <add key="WorkerCycleTime" value="10000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 95: Konfiguration Worker 2, WorkerPool: CadPool

```
<appSettings>
  <add key="WorkerNumber" value="3"/>
  <add key="WorkerPool" value="CadPool"/>
  <add key="WorkerCycleTime" value="10000"/>
  <add key="BatchSize" value="1"/>
</appSettings>
```

Abbildung 96: Konfiguration Worker 3, WorkerPool: CadPool