



Kundeninformation

Telekom **MMS**

P!ONE E-Rechnung

Schnittstellenbeschreibung



Impressum

V1, 11/2025

telekom-mms.com

Autor

Andreas Kortüm

Redaktion

Dany Gmey

Einleitung

Elektronische Rechnungen effizient versenden und empfangen – genau dafür sind unsere Lösungen **P!ONE E-Rechnung** und **P!ONE Peppol Access Point** gemacht.

Als Teil der Peppol-Infrastruktur hilft Ihnen unser Service P!ONE Peppol Access Point (PAP), Rechnungsprozesse zu digitalisieren und zu standardisieren – schnell, sicher und kompatibel mit aktuellen Vorgaben. Einführende Informationen erhalten Sie in unserem Dokument “Peppol Basics und technische Voraussetzungen”.

In diesem Dokument finden Sie alles, was Sie für die technische Anbindung an den P!ONE Peppol Access Point benötigen:

- Welche Schnittstellen stehen zur Verfügung?
- Wie erfolgt der Zugriff?
- Wie erfolgt die Authentifikation?
- Wie funktioniert der Versand über das Peppol-Netzwerk?
- Wie funktioniert der Empfang über das Peppol-Netzwerk?
- Wie erhalte ich Versandbestätigungen bzw. Fehlermeldungen?

Die Anleitung richtet sich an technische Projektverantwortliche und alle, die Schnittstellen einrichten oder in bestehende Systeme integrieren möchten. Ziel ist es, Ihnen eine klare, verständliche und praxisnahe Unterstützung an die Hand zu geben, damit Sie Ihre Anbindung reibungslos planen und umsetzen können.

Inhalt

Grundlagen der Schnittstellenarchitektur	1
1 4-Corner-Model	2
2 Anbindung an den P!ONE Peppol Access Point	4
2.1 Zulässige Dokumententypen	4
SFTP-Schnittstelle	6
3 SFTP-Schnittstelle im Überblick	7
4 Voraussetzungen	8
5 Zugriffsdaten	9
6 Authentifizierung testen	10
7 Dokumente versenden	11
8 Dokumente empfangen	13
REST-Schnittstelle	14
9 REST-Schnittstelle im Überblick	15
10 Voraussetzungen	16
11 Zugriffsdaten	17
12 Authentifizierung testen	18
13 Dokumente versenden	19
13.1 Validierungsfunktionen	20
13.2 Dokument absenden	21
14 Dokumente empfangen	22
14.1 Einsatz von WebHooks	23
Anhang	24
15 Erstellung eines SSH-Schlüssels mit PuTTYgen	25
15.1 Überblick SSH-Schlüssel	25

15.2	Schlüsseltool PuTTYgen	25
15.3	SSH-Schlüssel mit PuTTYgen erstellen	26
15.4	Privaten Schlüssel in OpenSSH konvertieren	28
16	Verarbeitungsquittung des Versanddienstes	29
16.1	Einfluss der verwendeten Schnittstelle	29
16.2	HTTP-Statuscodes und ihre Bedeutung	29
16.3	Datenstruktur der Verarbeitungsquittung	30
16.4	Feldbeschreibung	33
16.5	Liste der erwartbaren Fehlercodes	34
17	Glossar	37

Grundlagen der Schnittstellenarchi- tektur

1 4-Corner-Model

Peppol basiert auf dem 4-Corner Model (siehe Abbildung), bestehend aus Sender (Corner 1: C1), Empfänger (Corner 4: C4) und je einem Access Point auf beiden Seiten (Corner 2 & Corner 3: C2 & C3). Der Austausch von Nachrichten erfolgt:

1. Der Sender (C1) erstellt ein elektronisches Dokument und übermittelt es an seinen Peppol Access Point (C2).
2. Der Sender-Access Point (C2) validiert das Dokument und leitet es über das Peppol-Netzwerk an den Empfänger-Access Point (C3) weiter – adressiert über die Peppol-ID des Empfängers.
3. Der Empfänger-Access Point (C3) nimmt das Dokument entgegen, prüft die Zustellbarkeit und stellt sie dem Empfänger (C4) bereit.
4. Der Sender-Access Point (C2) benachrichtigt den Sender (C1) über die erfolgreiche Zustellung an den Empfänger-Access Point (C3).

Der Empfänger (C4) ruft das Dokument über seinen Peppol Access Point ab. Der gesamte Austausch erfolgt verschlüsselt, standardisiert und interoperabel, unabhängig davon, welche Systeme oder Anbieter die Beteiligten nutzen.

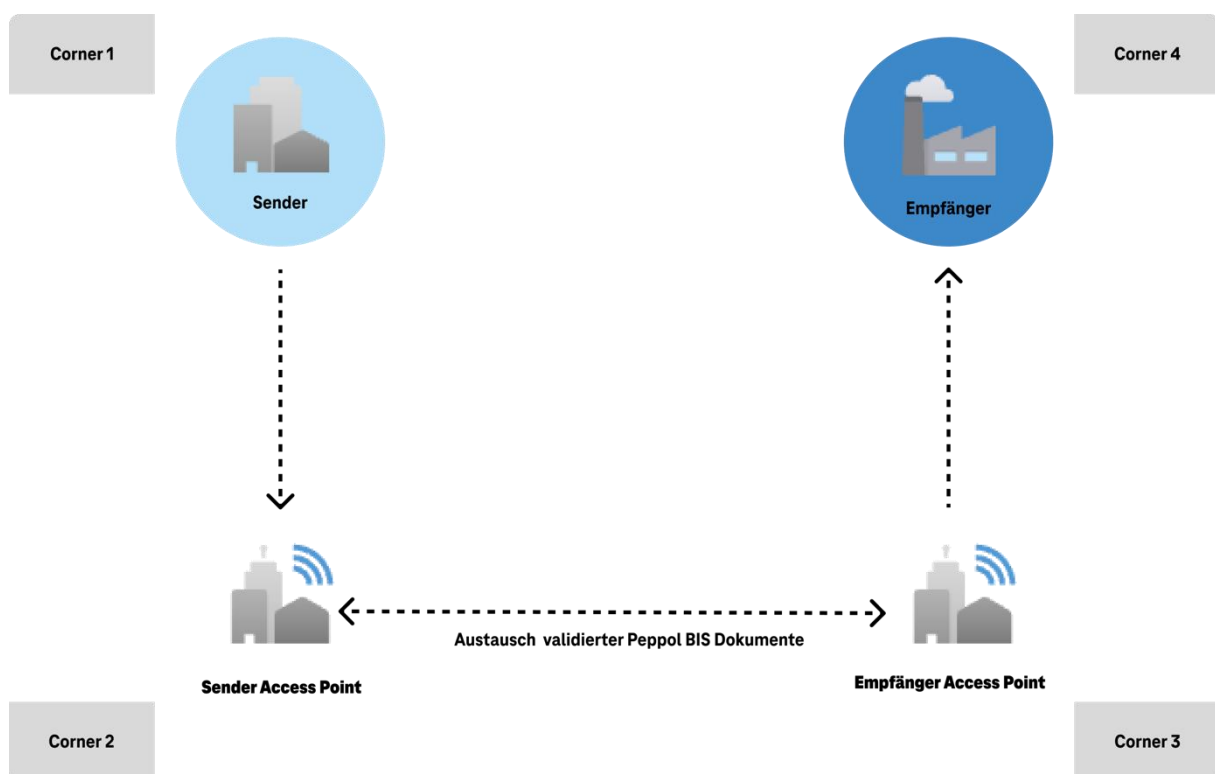


Abbildung 1: "4-Corner-Model"

Neben der reinen Nachrichtenübertragung standardisiert das Peppol-Netzwerk auch das Peppol Directory, in welchem Peppol-Teilnehmer festlegen, welche Dokumententypen Sie empfangen können¹. Jeder Peppol-Teilnehmer lässt diese Information durch seinen Access Point Provider

¹ Der Eintrag in das öffentlich einsehbare Peppol Directory ist empfohlen, aber nicht verpflichtend. Für den Peppol Service Provider kann diese Information allerdings über jeden Peppol-Teilnehmer abgefragt werden. Daher ist die technische Validierung des P!ONE Peppol Access Points vollständiger als die manuelle Prüfung über das Directory.

pflegen. Bei der Kommunikation wird dann gegen diese Informationen geprüft, was dem Netzwerk insgesamt eine höhere Zuverlässigkeit ermöglicht.

Diese Festlegung ist durch Peppol direkt einsehbar unter <https://directory.Peppol.eu/public>.

Die Nutzung der Kommunikationsstrecke zwischen den Peppol Access Points (C2-C3) ist durch den Netzverband Peppol.org standardisiert. Dies gilt nicht nur für das Dokumentenformat und das Übertragungsprotokoll, sondern auch für die Servicequalität, die Datensicherheit und das Logging aller Peppol_Access Points. Weitere Informationen dazu finden Sie unter:

<https://Peppol.org/learn-more/Peppol-interoperability-framework/>.

2 Anbindung an den P!ONE Peppol Access Point

Die Kommunikation innerhalb des Peppol-Netzwerkes, also die zwischen den **Peppol Access Points (PAP)** C3 und C4, ist streng normiert. Die Kommunikation mit den Endkunden (C1 bzw. C4) kann dagegen individuell zwischen dem Peppol Service Provider und dem Endkunden gestaltet werden.

Der P!ONE Peppol Access Point normiert diese Kommunikation und bietet zwei alternative Schnittstellen an, mit denen sich Endkunden einfach und zuverlässig anbinden lassen.

Mit dem **Secure File Transfer Protokoll (SFTP)** können Dateien sicher an den P!ONE Peppol Access Point übergeben bzw. von dort abgeholt werden. Dies ist prinzipiell auf Kundenseite sowohl manuell mit einem entsprechenden FTP-Client als auch per Scripting möglich. Die Initiative liegt beim Senden wie beim Empfangen auf Seiten des Clients (Kunden). Insbesondere für den Empfang ist es daher wichtig, in regelmäßigen Abständen zu prüfen, ob Dokumente eingegangen sind.

Über die **Representational State Transfer-Schnittstelle (REST)** kann mittels des HTTP-Protokolls auf Funktionen und Dokumente des P!ONE Peppol Access Points zugegriffen werden. Diese Schnittstelle wird aus einem vom Kunden zu programmierenden Programm heraus aufgerufen und ist im standardisierten Beschreibungsformat OpenAPI verfügbar. Wenn Sie sich für den programmierten Zugriff entscheiden, bietet diese Variante der Schnittstelle neben den Basisfunktionen, die analog zur SFTP-Schnittstelle sind, zusätzliche Funktionen z. B. zur Vorab-Validierung oder zur Benachrichtigung per WebHook.

Es kann entweder die PAP SFTP-Schnittstelle *oder* die PAP REST-Schnittstelle genutzt werden. Dies ist bei der Bestellung anzugeben.

2.1 Zulässige Dokumententypen

Das Peppol-Netzwerk lässt es nicht zu, beliebige Dokumente zu übertragen. Ein wichtiger Mehrwert besteht gerade in der strikten Einhaltung von Dokumentenstandards. Daher definiert Peppol eine Liste zulässiger Dokumententypen mit präziser Spezifikation.

Jeder Peppol Service Provider (C2/C3) wählt eine Untermenge dieser Dokumente aus, die er als gültig für den Empfang bzw. den Versand verarbeitet.

Jeder Endpunkt (=Endkunde mit eindeutiger Peppol-ID) des Peppol-Netzwerkes (C1/C4) wählt wiederum von dieser Menge eine Untermenge aus, die für ihn zulässig sein soll. Diese Informationen werden vom Peppol Service Provider im Netzwerk standardisiert geteilt.

Folgende Dokumententypen stehen durch den P!ONE Peppol Access Point zur Auswahl:

Dokumententyp	Zum Senden	Zum Empfangen
Peppol BIS Billing UBL Invoice	Aktuelle Version und Vorgängerversion	Aktuelle Version und Vorgängerversion
Peppol BIS Billing UBL Credit-Note		Aktuelle Version und Vorgängerversion

Peppol XRechnung UBL Invoice	Aktuelle Version und Vorgängerversion	Aktuelle Version und Vorgängerversion
Peppol XRechnung UBL Invoice Extension		Aktuelle Version und Vorgängerversion
Peppol XRechnung UBL Credit-Note		Aktuelle Version und Vorgängerversion

Tabelle 1: Zulässige Dokumententypen

Sollte das von ihnen benötigte Format fehlen, sprechen Sie uns gerne an.

SFTP-Schnittstelle

3 SFTP-Schnittstelle im Überblick

Das FTP-Protokoll ist ein normiertes Verfahren zur Übertragung von Dateien über das Internet. Ist dieses Protokoll zusätzlich gesichert, spricht man vom **SFTP-Protokoll** (Secure FTP). Das Protokoll selbst bietet eine kleine Anzahl einfacher Kommandos und kann über die Kommandozeile ausgeführt werden. Bei manueller Benutzung und für Testzwecke ist die Verwendung eines SFTP-Clients mit Benutzeroberfläche wesentlich komfortabler.

Weitere Informationen zum FTP-Protokoll und möglichen FTP-Client-Programmen finden Sie im Glossar am Ende dieses Dokuments. Unsere Qualitätssicherung nutzt beispielsweise den FTP-Client „FileZilla“ (Open Source).

Wenn Sie den **P!ONE Peppol Access Point** beauftragen möchten, teilen Sie uns bitte die gewünschte Schnittstelle mit. Mit diesem Auftrag legen wir Ihnen einen eigenen Mandanten an. Sobald Ihr SFTP-Mandant auf dem P!ONE Peppol Access Point angelegt ist, erhalten Sie von unserem Service Team (peppol-support@telekom-mms.com) Ihre Mandantenkennung. In dieser Dokumentation wird die Beispiel-Mandantenkennung „mustermann_gmbh“ verwendet.

Der P!ONE Peppol Access Point ist unter folgenden **URL-Adressen** erreichbar:

- Die **Testumgebung** ist erreichbar unter epsstgpapsto001.blob.core.windows.net.
- Die **Produktivumgebung** ist erreichbar unter epsprdpapsto001.blob.core.windows.net.

Ein direkter Zugriff auf diese Internetadressen ist *nicht* möglich, da sie durch **Sicherheitszertifikate (SSH-Schlüssel)** vor nicht autorisiertem Zugriff geschützt sind. Diese Schlüssel bestehen aus einem **privaten** und einem **öffentlichen Bestandteil**. Jeder Kunde erzeugt sich lokal einen individuellen Schlüssel. Der **private Bestandteil dieses Schlüssels ist nur Ihnen bekannt**, den **öffentlichen Bestandteil** übergeben Sie unserem P!ONE Service Desk. Mit dem international standardisierten SSH-Schlüssel-Verfahren wird sichergestellt, dass nur Sie auf ihre Postfächer zugreifen können.

Für den Fall, dass Ihnen das SSH-Schlüssel-Verfahren nicht vertraut ist, haben wir eine kleine [Anleitung](#) zur Erstellung dieses Schlüssels unter Windows erstellt. Diese finden Sie im Anhang.

4 Voraussetzungen

In der folgenden Übersicht finden Sie noch einmal alle Voraussetzungen:

1. Registrierung und Vertrag

Sie haben sich für den P!ONE Peppol Access Point mit SFTP-Schnittstelle registriert und einen gültigen Nutzungsvertrag abgeschlossen.

2. SFTP-Client

Sie haben einen SFTP-Client installiert bzw. zur Verfügung.

3. SSH-Schlüssel

- Sie haben einen SSH-Schlüssel erzeugt.
- Sie haben den **privaten Teil** des SSH-Schlüssels **in Ihrem SFTP-Client hinterlegt** und Sie haben den **öffentlichen Teil** des SSH-Schlüssels an unseren P!ONE Service Desk **übergeben**.

4. Mandantenkennung

Sie erhalten daraufhin vom P!ONE Service Desk Ihre Mandantenkennung und können starten.

5 Zugriffsdaten

Zum Zugriff brauchen Sie folgende Eckdaten:

URL der Testumgebung	epsstgpapsto001.blob.core.windows.net
URL der Produktivumgebung	epsprdpapsto001.blob.core.windows.net
Port	22
User	Wird Ihnen vom P!ONE Service Team zugeschickt
Verzeichnis zum Senden	OUTBOX
Verzeichnis zum Empfangen	INBOX
Verzeichnis für Versandprotokoll	INBOX
Frequenz für das Versenden	Alle 5 Minuten
Frequenz für das Empfangen	Alle 5 Minuten
Zustellverfahren bei Problemen	Die Zustellversuche werden bei <i>temporär nicht erreichbaren Empfängern</i> automatisch bis zu 15-mal im Abstand von jeweils einer halben Stunde wiederholt.

Tabelle 2: Zugriffsdaten SFTP-Schnittstelle

6 Authentifizierung testen

Beim erstmaligen Zugriff sollten Sie das Authentifizierungsverfahren testen. Grundlage der folgenden Schritte ist, dass die Voraussetzungen (Kapitel 4) vollständig erfüllt sind.

Dazu gehen Sie wie folgt vor:

1. Starten Sie Ihren SFTP-Client – in unserem Beispiel das Programm FileZilla
2. Geben Sie den Server (Testsystem) an: `epsstgpapsto001.blob.core.windows.net`
3. Geben Sie den Port an: 22
4. Geben Sie Ihren User an. Die User ID ist Ihre Mandantenkennung.
5. Es ist kein Passwort erforderlich, diese Funktion übernimmt der private Key.
6. Klicken Sie „Verbinden“
7. Wenn Sie erstmals auf unseren Server zugreifen, ist dieser Server in Ihrem System noch nicht bekannt. Daher werden Sie explizit gefragt, ob Sie diesem Server vertrauen. Bitte bestätigen Sie das. Es ist empfehlenswert, hier die Option „Diesem Host immer vertrauen“ anzuwählen, damit Sie bei künftigen Aufrufen unseres Servers nicht erneut gefragt werden.

Die Verbindung sollte nun erfolgreich bestätigt werden.

Falls die Verbindung nicht erfolgreich war, gibt Ihnen Ihr SFTP-Client einen entsprechenden Fehler aus. Die folgende Tabelle hilft bei der Identifikation der Fehlerursache:

Meldung des SFTP-Clients	Problembehebung
ssh_init: Host does not exist	Die Ziel-URL des SFTP-Servers ist ungültig. Bitte überprüfen Sie die korrekte Schreibweise.
Zeitüberschreitung der Verbindung nach 20 Sekunden Inaktivität Herstellen der Verbindung zum Server fehlgeschlagen	Die URL ist richtig, aber der SFTP-Server antwortet nicht. Überprüfen Sie die Korrektheit des Ports.
Herstellen der Verbindung zum Server fehlgeschlagen	Ihr User ist nicht gültig

Tabelle 3: Übersicht Authentifizierungsfehler SFTP-Schnittstelle

Führen Sie den gleichen Verbindungstest auch mit dem Produktivserver durch. Idealerweise speichern Sie Test und Produktion als Serverprofile in Ihrem SFTP-Client ab. Damit können Sie diese fortan zuverlässig nutzen.

7 Dokumente versenden

Um ein Dokument zu versenden, stellen Sie mit dem zuvor erläuterten Autorisierungsverfahren eine SFTP-Verbindung her. Dann wechseln Sie in das Verzeichnis OUTBOX und legen dort die zu versendende Datei ab. Nach max. 5 Minuten verschwindet diese Datei. Passend zu dieser Datei erhalten Sie eine Protokolldatei. Laden Sie die Protokolldatei herunter und schauen Sie einmal rein, um sicher zu sein, dass keine Fehler aufgetreten sind. Anschließend beenden Sie diese Verbindung.

Der Ablauf sieht im Detail wie folgt aus:

Ablauf Versand

1. Sie starten Ihren SFTP-Client
2. Sie stellen die SFTP-Verbindung zum P!ONE Peppol Access Point her
3. Sie haben eine Zieladresse – das ist die Peppol ID des Empfängers
4. Sie haben eine Datei, die einem gültigen Peppol-Format (siehe Kapitel 2.1) entspricht
5. Das gewählte Format entspricht dem Format, das der Empfänger auch empfangen möchte
6. Die Zieladresse (Peppol-ID des Empfängers) und Ihre Absenderadresse (Ihre Peppol-ID) sind in der zu senden Datei enthalten
7. Sie wechseln in das Verzeichnis OUTBOX (SFTP-Kommando CD)
8. Sie legen Ihre Datei im OUTBOX-Verzeichnis ab (SFTP-Kommando PUT).
9. Der P!ONE Peppol-Access holt die Datei automatisch ab und prüft folgende Kriterien:
 - Syntaktische Korrektheit der eingegangenen Datei
 - Identifizierung des Empfängers im Peppol-Netzwerk
 - Prüfen der vom Empfänger akzeptierten Dokumententypen und -versionen
 - Bei einem Fehler überprüfen Sie bitte die Punkte 3 bis 5
10. Die Datei wird immer aus der OUTBOX entfernt, sobald sie verarbeitet wurde.
11. Sofern die Validierung fehlerfrei absolviert wurde, wird das Dokument über das Peppol-Netzwerk an den Access Point des Empfängers weitergeleitet.
12. Ein Dokument gilt als erfolgreich übergeben, sobald es erfolgreich auf dem C3 -Server (das ist der Peppol Access Point des Empfängers) abgelegt werden konnte.
13. Rückmeldungen oder technische Bestätigungen (Protokolldatei) landen in ihrer Inbox. Lassen Sie sich dazu den Inhalt der INBOX anzeigen (SFTP-Kommando LS). Laden Sie sich diese Datei herunter (SFTP-Kommando GET) und prüfen Sie, ob die Übertragung fehlerfrei bestätigt wurde. Der Dateiname der Protokolldatei entspricht dem Dateinamen Ihrer Originaldatei, ergänzt um den Namenszusatz „.out“.
14. Wenn Sie die Protokolldatei heruntergeladen haben, sollten Sie sie aus der INBOX entfernen, damit Sie den Überblick behalten (SFTP-Kommando RM).
15. Sie melden sich vom P!ONE Peppol Access Point ab.

Sollten Fehler aufgetreten sein, finden Sie eine Erläuterung dazu in Kapitel 16.

Eine positive Protokolldatei bedeutet, dass das Dokument fehlerfrei im **Peppol-Netzwerk** von C2 an C3 übertragen wurde. Ob es vom Empfänger (Corner 4: C4) tatsächlich abgeholt wurde und ob die Rechnung auch inhaltlich akzeptiert wurde, geht aus dieser Protokolldatei *nicht* hervor.

Die Peppol-Kommunikation sieht dafür ein eigenes Dokument INVOICE RESPONSE vor. Leider ist diese zweite Bestätigung noch wenig verbreitet. Die Spezifikation des Verfahrens finden Sie unter <https://docs.Peppol.eu/poacc/upgrade-3/profiles/63-invoiceresponse/>.

Der Versanddienst unseres Peppol Access Points stellt das Ergebnis jeder Dokumentenverarbeitung als XML-Datei zur Verfügung. Details werden in Kapitel 16 beschrieben. Damit können Schnittstellenentwickler und Integrationsverantwortliche frühzeitig planen, wie die Rückgabedaten systemseitig genutzt und verarbeitet werden.

8 Dokumente empfangen

Ein Partner hat Ihnen mitgeteilt, dass er **per Peppol** ein Dokument erfolgreich an Sie verschickt hat. Um ein Dokument zu empfangen, stellen sie mit dem zuvor erläuterten Autorisierungsverfahren eine SFTP-Verbindung her. Dann wechseln Sie in das Verzeichnis INBOX und prüfen, ob dort die erwartete Datei abgelegt ist. Wenn keine Prüfungen fehlgeschlagen sind, sollte diese Datei direkt verfügbar sein. Sie laden diese Datei lokal herunter und löschen Sie in der INBOX. Anschließend beenden Sie die Verbindung.

Was Sie im Fehlerfall tun können, wird am Ende dieses Abschnitts erklärt.

Der Ablauf sieht im Detail wie folgt aus:

Ablauf Empfang

1. Sie starten Ihren SFTP-Client.
2. Sie stellen die SFTP-Verbindung zum P!ONE Peppol Access Point her.
3. Sie wechseln in das Verzeichnis INBOX (SFTP-Kommando CD).
4. Sie lassen sich den Inhalt des Verzeichnisses INBOX anzeigen (SFTP-Kommando LS).
5. Sie (C4, Rechnungsempfänger) prüfen Ihr INBOX-Verzeichnis auf neue Dateien.
6. Sie finden eine neue Datei und laden diese lokal herunter (SFTP-Kommando: GET).
7. Sie entfernen die heruntergeladene Datei aus der INBOX (SFTP-Kommando: RM), damit Sie nicht versehentlich bereits erhaltene Dateien als neu empfangene Dateien interpretieren.
8. Sie melden sich vom P!ONE Peppol Access Point ab.

Der Empfang von Dokumenten erfolgt jederzeit und unabhängig davon, ob Sie sich einwählen oder nicht. Es werden in der INBOX alle an Sie gesendeten Dokumente aufbewahrt. Wir empfehlen, diese Dateien innerhalb von 1 Woche abzuholen und aus der INBOX zu entfernen. Die automatische Löschung von Dateien richtet sich nach der Mengenüberwachung (Quota) und kann variieren. Die Dateien bleiben aber mindestens 1 Monat in der INBOX.

Der P!ONE Peppol-Access Point (C3) prüft bei Empfang folgende Eigenschaften ab:

- Identifizierung des Empfängers im Peppol-Netzwerk
- Prüfen der vom Empfänger akzeptierten Dokumententypen und -versionen
- Syntaktische Korrektheit der eingegangenen Datei

REST-Schnittstelle

9 REST-Schnittstelle im Überblick

Über die **P!ONE Peppol Access Point REST-API** (PAP-REST) können Dokumente synchron an das **Peppol-Netzwerk** übermittelt werden. Die Zustellung erfolgt in Echtzeit, inklusive direkter Rückmeldung über den Erfolg oder mögliche Fehler beim Versand. Der Empfang von Dokumenten ist im Peppol-Netzwerk immer asynchron. PAP-REST bietet Ihnen zwei Wege, den Empfang zu automatisieren: Sie können zeitgesteuert auf neu eingegangene Dokumente prüfen oder sich durch einen WebHook über den Eingang eines neuen Dokumentes informieren lassen.

Dieses Kapitel ist eine Ergänzung der im JSON-Format bereitgestellten OPENAPI-Dokumentation in den Dateien

PAP openAPI senden V1_2_0.json

PAP openAPI empfangen V2_0_0.json

Sollte Ihre Entwicklungsumgebung keinen Viewer für OPENAPI-Dokumentation bereitstellen, können Sie entsprechende Viewer im Internet finden, beispielsweise unter <https://www.openapi-viewer.com/>.

Die OPENAPI-Dokumentation wird wie in der Softwareentwicklung üblich in englischer Sprache bereitgestellt.

10 Voraussetzungen

In der Übersicht hier noch einmal alle Voraussetzungen:

1. Registrierung und Vertrag

Sie haben sich für den P!ONE Peppol Access Point mit REST-Schnittstelle registriert und einen gültigen Nutzungsvertrag

2. Mandantenkennung

Sie haben vom P!ONE Service Desk Ihre Mandantenkennung erhalten

3. Aufruf REST-Schnittstelle

Sie haben ein Softwareprogramm, mit dem Sie über das Internet REST-Schnittstellen aufrufen können

4. TLS-Zertifikat

Sie haben uns ein gültiges TLS-Zertifikat bereitgestellt.

Wichtig: Wir benötigen von Ihnen ein TLS-Zertifikat mit folgenden Eigenschaften:

- Es ist von einer **autorisierten Zertifikatsstelle** erzeugt (aus Sicherheitsgründen können wir keine selbst erzeugten Zertifikate akzeptieren).
- Wildcard-Zertifikate werden akzeptiert.

Das TLS-Zertifikat hat eine begrenzte Laufzeit. Nach Ablauf der Gültigkeit wird es vom P!ONE Peppol Access Point nicht mehr akzeptiert. Bitte setzen Sie sich rechtzeitig vor Ablauf ihres Zertifikates mit uns in Verbindung.

Wenn Sie auf der Suche nach einem TLS-Zertifikat-Anbieter sind, empfehlen wir Ihnen den Zertifikatsservice der Deutschen Telekom unter: <https://www.telesec.de/de/loesungen/certificate-and-signing-services/ueberblick>

Wir nutzen das TLS-Zertifikat zur beidseitigen Absicherung der Kommunikationsverbindung („mutual TLS“). Entsprechend muss das TLS-Zertifikat auf eine von Ihrem Unternehmen im Internet sichtbare und mit TLS zertifizierte Server-URL ausgestellt sein.

11 Zugriffsdaten

Zum Zugriff brauchen Sie folgende Eckdaten:

URL der Testumgebung - senden	mci-stg-phb.procurement-one.de/connect
URL der Testumgebung - empfangen	mci-stg-phb.procurement-one.de/Peppol
URL der Produktivumgebung - senden	mci-phb.procurement-one.de/connect
URL der Produktivumgebung - empfangen	mci-phb.procurement-one.de/Peppol
Port	443 (HTTPS Standard)
User	Wird Ihnen vom P!ONE Service Team zugeschickt
Aufruf zum Senden	
Aufruf zum Empfangen	get /fs/v2/files
Frequenz für das Versenden	Alle 5 Minuten
Frequenz für das Empfangen	Alle 5 Minuten
Zustellverfahren bei Problemen	Die Zustellversuche werden bei <i>temporär nicht erreichbaren Empfängern</i> automatisch bis zu 15-mal im Abstand von jeweils einer halben Stunde wiederholt.

Tabelle 4: Zugriffsdaten REST-Schnittstelle

12 Authentifizierung testen

Beim erstmaligen Zugriff sollten Sie das Authentifizierungsverfahren testen. Ein Ping oder Tracert auf die Ziel-URL ist dabei nicht möglich, da der P!ONE Peppol Access Point das entsprechende Protokoll für diese Befehle aus Sicherheitsgründen nicht unterstützt.

Stellen Sie außerdem sicher, dass Ihre ausgehende Firewall die Ziel-Domain unter dem festgelegten Port erreichen darf.

Rufen Sie die URL des Testsystems mit dem HTTPS-Protokoll auf, beispielsweise als

```
Curl -vvv https://mci-stg-phb.procurement-one.de/connect
```

für den Versand von Dokumenten.

Die Verbindung sollte nun erfolgreich bestätigt werden.

Falls die Verbindung nicht erfolgreich war, hilft die folgende Tabelle bei der Identifikation der Fehlerursache:

Meldung	Problembehebung
401 Unauthorized	Der Aufruf erfolgte ohne TLS-Zertifikat.
tls: unknown certificate authority	Das verwendete TLS-Zertifikat wurde nicht von einem autorisierten Zertifikatsanbieter erstellt.
Tracert/Ping: Zeitüberschreitung der Anforderung.	Das ICMP-Protokoll für Tracert/Ping ist aus Sicherheitsgründen nicht freigeschaltet.
Failed to connect to mci-stg-phb.procurement-one.de port 443: Connection refused	Dieser Fehler tritt typischerweise auf, wenn eine Firewall den Zugriff verbietet. Entsprechend sind die Firewallregeln (Domain, Port, Direction) auf beiden Seiten zu prüfen.

Tabelle 5: Authentifizierungsfehler REST-Schnittstelle

Führen Sie den gleichen Verbindungstest auch mit dem Produktivserver durch. Idealerweise speichern Sie Test und Produktion als Serverprofile ab, sodass Sie diese fortan zuverlässig nutzen können.

13 Dokumente versenden

Über die REST-API können strukturierte Rechnungsdokumente synchron an das Peppol-Netzwerk übermittelt werden. Die Zustellung erfolgt in Echtzeit an C3, inklusive direkter Rückmeldung über den Erfolg oder mögliche Fehler beim Versand. Ob die Zustellung von C3 nach C4 synchron oder asynchron erfolgt, liegt im Ermessen des Empfängers (C3/C4). Ihr System sendet strukturierte Peppol-Dokumente direkt per HTTPS-POST an die REST-API. Die HTTPS-Antwort enthält unmittelbar die verbindliche Quittung des Empfänger-Access Points (Erfolg oder Fehler).

Ablauf Versand

1. Ihr System (C1, Rechnungssender) sendet ein Dokument per `post/api/v1/send` (oder einer der Variationen von `send`)
2. Der P!ONE PAP (C2) prüft automatisch auf folgende Kriterien:
 - Syntaktische Korrektheit der eingegangenen Datei
 - Identifizierung des Empfängers im Peppol-Netzwerk
 - Prüfen der vom Empfänger akzeptierten Dokumententypen und -versionen
 - Bei einem Fehler wird das Dokument gelöscht und nicht verschickt. Die gefundenen Fehler werden in der Verarbeitungsquittung (Rückgabeparameter) aufgeführt.
3. Nach erfolgreicher Prüfung wird das Dokument über das Peppol-Netzwerk an den Access Point des Empfängers (C3) weitergeleitet.
4. Der P!ONE PAP wartet synchron auf die Übertragungsbestätigung von C3.
5. C3 bestätigt die erfolgreiche Übertragung an C2.
6. Die Bestätigung (=Verarbeitungsquittung) wird als Rückgabeobjekt der Funktion zurückgegeben.

Es wird deutlich, dass C2 lediglich den technischen Übertragungsstatus an Corner 3 (C3) verlässlich bestätigen kann. Ob das Dokument von C3 erfolgreich an den finalen Empfänger (Corner 4: C4) ausgeliefert wurde, kann C2 *nicht* übermitteln.

Fachliche Informationen zur weiteren Verarbeitung auf Empfängerseite (C4) können über spezifische Nachrichtentypen innerhalb des Peppol-Netzwerks bereitgestellt werden. Diese stellen jeweils eine eigenständige Peppol-Kommunikation dar.

Verarbeitungsergebnisse des P!ONE Versanddienstes

Der Versanddienst unseres Peppol Access Points stellt das Ergebnis jeder Dokumentenzustellung wahlweise im XML- oder JSON-Format zur Verfügung. Die Auswahl erfolgt per Aufrufparameter (siehe OpenAPI-Dokumentation). Die Rückgabedaten werden direkt an den aufrufenden Client übermittelt. Details sind in Anhang 16 beschrieben. Dadurch können Schnittstellenentwickler und Integrationsverantwortliche frühzeitig planen, wie die Rückgabedaten systemseitig genutzt und verarbeitet werden.

13.1 Validierungsfunktionen

Die API bietet die Möglichkeit, die beim Senden stets durchgeführten Prüfungen auch einzeln aufzurufen. Dadurch kann das aufrufende Programm detailliert auf Validierungsprobleme reagieren und die Chance auf einen erfolgreichen Versand deutlich erhöhen.

Die angebotenen Validierungsmöglichkeiten umfassen:

- **Gültige Peppol-ID**

Sie können vorab prüfen, ob die Empfänger-ID (Peppol-ID) gültig ist.

Dazu nutzen Sie die Funktion `checkParticipant(PeppolId)` und werten den Status der Response aus. Nur der Status SUCCESS sollte akzeptiert werden.

Ist die Peppol-ID ungültig, wird der Status FAIL zurückgegeben.

- **Akzeptierte Dokumententypen**

Jeder Empfänger im Peppol-Netzwerk deklariert, welche Dokumententypen er akzeptiert. Sie können mit der Funktion `checkParticipant(PeppolId)` wahlweise die Liste der akzeptierten Dokumententypen auslesen (Response `documentTypes`) oder gezielt prüfen, ob der Dokumententyp, den Sie versenden möchten, von diesem Empfänger akzeptiert wird.

Wenn Sie beispielsweise eine E-Rechnung im Format „Peppol BIS Billing UBL Invoice V3“ an den Peppol-Empfänger „Telekom MMS“ versenden möchten, gehen Sie wie folgt vor:

1. Unter <https://directory.Peppol.eu/public> suchen Sie nach „Telekom MMS“. Sie erhalten die Auskunft, dass die Peppol-ID dieses Empfängers wie folgt lautet: 9930:de811807949_test
2. Unter <https://docs.Peppol.eu/edelivery/codelists/> suchen Sie für die aktuelle Version die Liste aller Document Types. Dort finden Sie unter „Peppol BIS Billing UBL Invoice V3“ als „Peppol Document Type Identifier Value“:

```
urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:cen.eu:en16931:2017#compliant#urn:fdc:Peppol.eu:2017:poacc:billing:3.0::2.1
```

3. Nun können Sie gezielt die Prüfung aufrufen:

```
checkParticipant(9930:de811807949_test, urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:cen.eu:en16931:2017#compliant#urn:fdc:Peppol.eu:2017:poacc:billing:3.0::2.1)
```

Auch hier muss das erwartete Ergebnis den Status SUCCESS zurückgeben.

- **Gültiges Dokument**

Über das Peppol-Netzwerk dürfen nur syntaktisch korrekte Dokumente verschickt werden. Beim Versand wird daher jedes Dokument syntaktisch geprüft und nur bei erfolgreicher Prüfung verschickt. Auch hier kann die Validierung vorgezogen werden, um programmatisch auf mögliche Fehlerfälle reagieren zu können.

Dazu sind folgende Möglichkeiten verfügbar:

Validate(Document)

Diese Funktion prüft das übergebene Dokument auf syntaktische Korrektheit. Dabei wird das Validierungsschema des Dokumententyps benutzt, der im Dokument selbst angegeben ist.

Alternativ können Sie den Dokumententyp auch explizit beim Aufruf angeben und damit genau auf diesen Dokumententyp validieren lassen. Der Aufruf ist dann

Validate(documentType, document)

Das Response-Objekt (PoneResponse) der Funktion muss den Status SUCCESS aufweisen, wenn das Dokument erfolgreich verschickt werden soll. Ist der Status nicht SUCCESS, enthält das Response-Objekt ein oder mehrere Messages, die über die gefundenen Fehler informieren.

Wird das zu prüfende Dokument als **Multipart** übertragen, gibt es dazu eine eigene Funktion: **validateMultipart()**.

Die Funktionsweise entspricht dabei der bereits erläuterten Validierung. Für die Übertragung eines Dokumentes im Modus „Binary“ gibt es derzeit keine gesonderte Validierung.

Hinweis: Bei der Dokumentenprüfung findet keine Prüfung der Peppol-ID des Empfängers oder der vom Empfänger akzeptierten Dokumententypen statt.

13.2 Dokument absenden

Ein Dokument kann auf drei verschiedene Arten gesendet werden: direkt als XML, als Multipart oder als Binary. Alle drei Alternativen haben die gleichen Parameter. Pflichtparameter ist dabei nur das Dokument selbst. Sender und Empfänger sind optional und werden, falls sie nicht explizit angegeben werden, aus dem Dokument selbst extrahiert.

Mit jedem Aufruf wird stets nur ein Dokument übergeben.

Das Response-Objekt ist analog zur Validierung aufgebaut. Nur wenn der Status SUCCESS lautet, wurde das Dokument erfolgreich verschickt. Im Fehlerfall wird das Dokument nicht verschickt und auch nicht gespeichert, sondern vollständig gelöscht.

Die folgende Tabelle gibt einen Überblick über die verfügbaren Funktionen der empfangenden PAP REST-API. Passend zur OpenAPI-Dokumentation ist diese Tabelle in Englisch verfasst.

Aktion	API-Funktion	Parameter	Returns
Check Participant	get /api/v1/checkParticipant	PeppolId, Accept (list of document types)	ParticipantInfo
Validate	post /api/v1/validate	documentType, processId, document	PoneResponse
ValidateMultipart	post /api/v1/validateMultipart	documentType, processId, document	PoneResponse
Send	post /api/v1/send	receiverId, senderId, document	PoneResponse
SendMultipart	post /api/v1/sendMultipart	receiverId, senderId, document	PoneResponse
SendBinary	post /api/v1/sendBinary	receiverId, senderId, document	PoneResponse

Tabelle 6: Funktionen der empfangenden PAP REST-API -Senden

14 Dokumente empfangen

Auf Seiten des fachlichen Empfängers (Corner 4: C4, Sie) übernimmt der empfangende Access Point (Corner 3: C3, wir) folgende Aufgaben:

- Empfang des Dokuments über das Peppol-Netzwerk
- Validierung des Dokuments
- Aktive Prüfung auf neue Nachrichten durch C4 mit der Funktion „list files“, alternativ Verwendung des WebHook-Verfahrens und aktive Benachrichtigung von C4
- Download der neuen Nachrichten
- Aktives Löschen der Nachrichten nach erfolgreichem Download

Rückgabedaten des Empfangsdienstes

Bei der Kommunikation unseres Empfangsdienstes mit dem technischen System des Kunden muss klar geregelt sein, wann beide Seiten (C3 und O4) die Übermittlung als erfolgreich bewerten. Daher muss bei der REST-API von Ihnen (C4) aktiv bestätigt werden, dass Sie ein Dokument erfolgreich heruntergeladen haben. Diese Bestätigung erfolgt, indem Ihr Client das Dokument über die API auf dem Fileserver löscht. Erst nach dieser Löschung entfernt unser Fileserver das Dokument aus seinem Speicher.

Antwort- und Fehlerdatenstrukturen

Die Fileserver-API stellt für jede mögliche API-Operation (z. B. Auflisten empfangener Dokumente, Abruf eines Dokuments) passende Antwortdatenstrukturen bereit. Die Fehlerdatenstruktur ist in allen Fällen gleich aufgebaut, allerdings einfacher als beim Versanddienst. Sie enthält:

- den passenden HTTP-Response-Status,
- einen Fehlercode als sprechende Zeichenkette (z. B. `bad_parameter`),
- einen Zeitstempel,
- eine Liste von Fehlermeldungen, die zum Fehlercode passen und Details liefern (Beispiel: Input parameter 'receivers' contains addresses which are not registered for this API. Unregistered addresses: 9930:de123456789).

Die Beschreibung der Fehlerdatenstruktur sowie die Wertemenge der Fehlercodes finden Sie in Kapitel 16.

Die folgende Tabelle gibt einen Überblick zu den verfügbaren Funktionen der empfangenden PAP REST-API. Passend zur OpenAPI-Dokumentation ist diese Tabelle in Englisch gehalten.

Aktion	API-Funktion	Parameter	Returns
List the files available for download	<code>get /fs/v2/files</code>	<code>maxResult</code> (Range 1 – 500) <code>receivers</code> (comma-separated list of <i>Peppol</i> -IDs)	List of unique file ids File metadata
Provides metadata of a specific file	<code>get /fs/v2/files/{fileId}/info</code>	<code>fileId</code> (Unique file id)	File metadata
Download of a specific file	<code>get /fs/v2/files/{fileId}</code>	<code>fileId</code> (Unique file id)	Binary of document / Content-Disposition
Confirms the successful	<code>delete /fs/v2/files/{fileId}</code>	<code>fileId</code> (Unique file id)	None

transfer of a specific file			
Download of a compressed file archive	get /fs/v2/nextArchive	receivers (comma-separated list of <u>Peppol</u> -IDs)	String Value for Iteration Content-Disposition
Confirms the successful transfer of several files	delete /fs/v2/files	Files (list of unique file ids)	None

Tabelle 7: Funktionen der empfangenden PAP REST-API - Empfang

Der Empfang von Dokumenten erfolgt jederzeit und unabhängig davon, ob Sie die API aufrufen oder nicht. Es werden auf unserem Server alle an Sie gesendeten Dokumente aufbewahrt. Wir empfehlen, diese Dateien innerhalb von 1 Woche abzuholen und per delete-Befehl zu entfernen. Die automatische Löschung von Dateien richtet sich nach der Mengenüberwachung (Quota) und kann variieren. Die Dateien bleiben aber mindestens 1 Monat verfügbar.

14.1 Einsatz von WebHooks

Wenn Sie den P!ONE Peppol Access Point nicht zyklisch auf neu eingegangene Dokumente prüfen wollen, können Sie alternativ auch das **WebHook-Verfahren** nutzen. Dazu stellen Sie Ihrerseits eine Aufruf-URL bereit, die im Falle einer eingehenden Nachricht vom P!ONE Peppol Access Point aufgerufen werden soll.

Die erwartete Konfiguration dieses Webhooks finden Sie in der OpenAPI-Dokumentation.

Die hinterlegte, aktuelle Konfiguration des Webhooks kann über die API-Funktion

```
get /fs/v2/webhook/fileReceiptEvent
```

abgerufen werden.

Da Sie der Bereitsteller des aufzurufenden Servers sind, legen Sie die konkreten Parameter für ihren WebHook fest. Entsprechend können Sie die Konfiguration mit folgender API-Funktion setzen bzw. ändern:

```
post /fs/v2/webhook/fileReceiptEvent
```

Wenn Sie sich entscheiden, das WebHook-Verfahren nicht weiter zu verwenden, löschen Sie die Konfiguration mit folgender API-Funktion:

```
delete /fs/v2/webhook/fileReceiptEvent
```

Wenn Sie einen Event erhalten haben, müssen Sie auf Ihrer Seite entsprechend reagieren. Ein Beispiel für eine Reaktion auf einen „Nachricht erhalten“ Event finden Sie unter:

```
post /fs/v2/webhook/fileReceiptEvent/onEvent
```

Anhang

15 Erstellung eines SSH-Schlüssels mit PuTTYgen

15.1 Überblick SSH-Schlüssel

Die Erstellung eines SSH-Schlüssels ist für die Nutzung der SFTP-Schnittstelle zur sicheren Verbindung erforderlich. Für die REST-Schnittstelle ist dieser Schlüssel nicht erforderlich.

Die Authentifizierungsmethode, basierend auf SSH-Keys (auch als Public-Key-Authentifizierung bekannt), bietet bedeutende Vorteile gegenüber der traditionellen Passwort-Anmeldung. Anstatt das Passwort oder dessen Hash-Wert auf dem Server zu speichern, wird lediglich der öffentliche Schlüssel hinterlegt. Der private SSH-Key bleibt auf Ihrem eigenen Gerät und darf nicht weitergegeben werden.

Zusätzlich besteht die Möglichkeit, den privaten Schlüssel mit einer Kennung zu verschlüsseln. Der öffentliche Schlüssel hat die Fähigkeit, Signaturen zu verifizieren, die mit dem privaten SSH-Schlüssel erstellt wurden, was eine nahtlose automatische Anmeldung auf dem Server ermöglicht. Selbst wenn Unbefugte Zugriff auf den öffentlichen Schlüssel erhalten, ist es ihnen nahezu unmöglich, den zugehörigen privaten Schlüssel zu berechnen.

Durch die Verwendung von Schlüssel-Paaren anstelle eines einfachen Passworts profitieren Sie nicht nur von einem vereinfachten Anmeldeprozess, sondern erhöhen auch die Sicherheit Ihrer Verbindung erheblich.

15.2 Schlüsseltool PuTTYgen

Eine Möglichkeit zur Erzeugung eines SSH-Schlüssels ist die Verwendung des Schlüsseltools „PuTTYgen“. Dieses Werkzeug ist eine eigenständige Software und nicht Teil der von der Telekom bereitgestellten Leistung. Diese Anleitung dient lediglich als anschauliches Beispiel, wie Sie ein SSH-Schlüssel erzeugen können. Alle Informationen zu diesem Tool inkl. Nutzungsbedingungen finden Sie unter:

<https://puttygen.com>

Diese Anleitung führt Sie Schritt für Schritt durch den Prozess der Erstellung eines SSH-Schlüssels mit PuTTYgen. Die Screenshots entstammen der Version 0.77-1 unter x86 Windows und können von Ihrer Installation abweichen. Selbstverständlich können Sie auch ein anderes Tool verwenden. Wir benötigen von Ihnen einen gültigen Schlüssel im OpenSSH-Format.

Folgende Schlüssel sind mit unserer Lösung kompatibel:

- RSA-SHA2-256
- RSA-SHA2-512
- ECDSA-SHA2-NISTP256
- ECDSA-SHA2-NISTP384
- ECDSA-SHA2-NISTP521

RSA-Schlüssel müssen mindestens 3.072 Bit lang sein.

15.3 SSH-Schlüssel mit PuTTYgen erstellen

Mit PuTTYgen können Sie unter Windows ganz einfach ein neues Schlüsselpaar für sichere Verbindungen erstellen. So gehen Sie Schritt für Schritt vor:

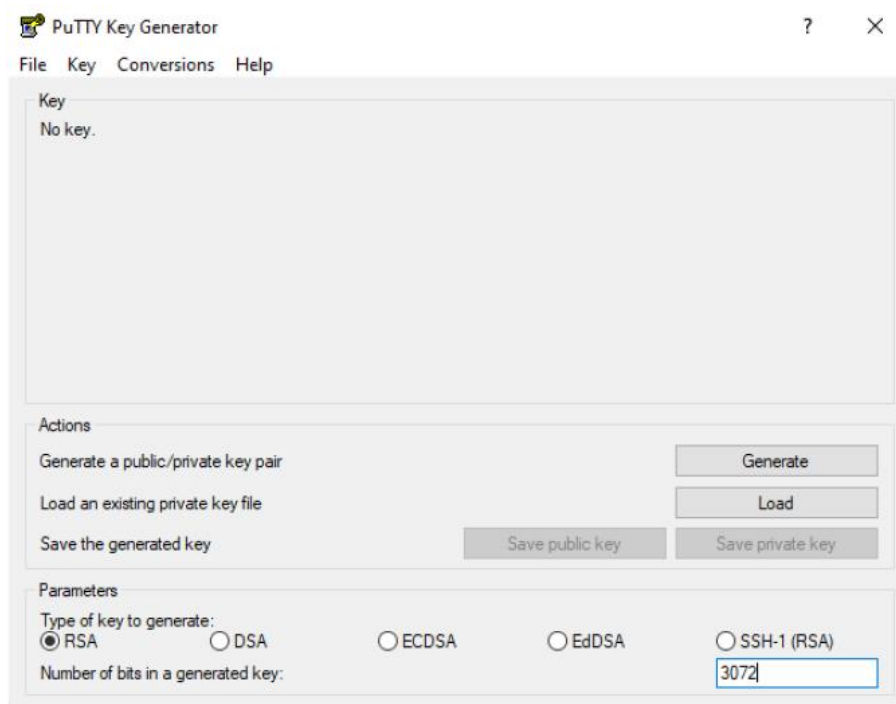
1. PuTTYgen starten

Öffnen Sie das Programm PuTTYgen. Sie finden es z.B. über das Startmenü oder in Ihrem PuTTY-Installationsordner.

2. Schlüsseltyp auswählen

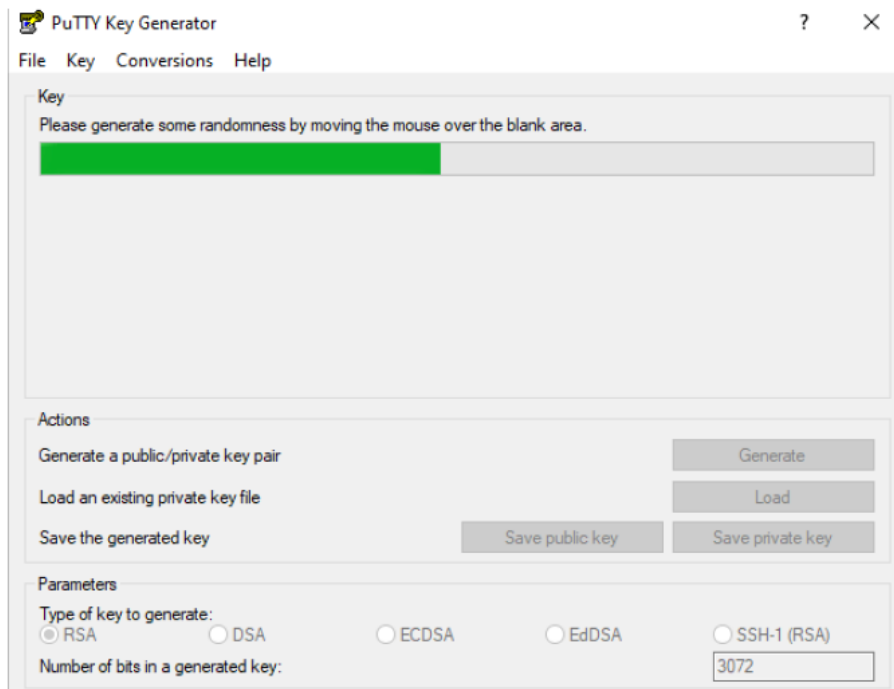
Unten im Fenster wählen Sie den gewünschten Schlüsseltyp aus. Empfohlene Optionen sind:

- RSA-SHA2-* (empfohlen, 3.072 Bit)
- ECDSA (Alternative mit kürzeren Schlüsseln, z. B. 256 Bit)



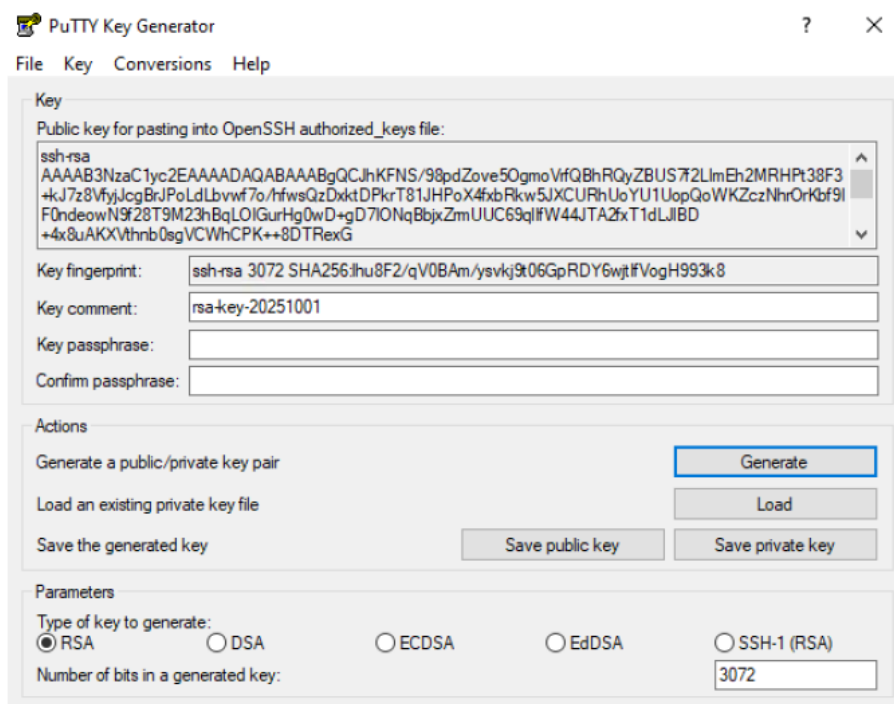
3. Schlüssel erzeugen

Klicken Sie auf "Generate". Nun beginnt die eigentliche Zufallsdatensammlung: Bewegen Sie die Maus zufällig im Fensterbereich. Je mehr Bewegung, desto besser. PuTTYgen nutzt diese Mausbewegungen, um einen sicheren, nicht vorhersehbaren Schlüssel zu erstellen.



4. Schlüssel fertigstellen

Sobald der grüne Balken voll ist, berechnet das Tool Ihren Schlüssel. Das dauert wenige Sekunden. Danach wird Ihnen der öffentliche Schlüssel oben im Fenster angezeigt.



5. Passphrase setzen

Tragen Sie nun eine Passphrase (eine lange, geheime Zeichenfolge) in die beiden Felder ein, um Ihren privaten Schlüssel zu schützen. Diese schützt den Schlüssel zusätzlich. Wenn Sie den Schlüssel automatisiert nutzen wollen, können Sie dieses Feld frei lassen.

6. Schlüssel speichern

Klicken Sie auf "Save private key", um Ihren privaten Schlüssel zu speichern (z. B. als id_rsa.ppk).

Wichtig: Diesen Schlüssel sicher aufbewahren - er ist vertraulich.

Speichern Sie nun den öffentlichen Schlüssel mit "Save public key".

7. Schlüssel verwenden

Kopieren Sie den öffentlichen Schlüssel aus dem Textfeld oben (per Copy & Paste) und fügen Sie ihn in Ihr GitHub- oder Serverprofil ein. Wenn Sie den Schlüssel später nutzen wollen, öffnen Sie PuTTY oder WinSCP und wählen Sie in den Verbindungseinstellungen Ihren .ppk-Schlüssel aus (dieses Format ist speziell für PuTTY gedacht. Für andere Anwendungen können Sie den Schlüssel ggf. in ein OpenSSH-Format konvertieren.

15.4 Privaten Schlüssel in OpenSSH konvertieren

Wenn Sie den privaten Schlüssel für OpenSSH (Linux/macOS, Git Bash etc.) brauchen, können Sie ihn aus dem .ppk-Format exportieren. Die Umwandlung betrifft nur den privaten Schlüssel. Den öffentlichen Teil können Sie weiterhin aus PuTTYgen kopieren oder aus dem privaten Schlüssel ableiten.

So funktioniert es:

1. PuTTYgen öffnen (sofern Sie die Anwendung bereits geschlossen haben)
2. Klicken Sie auf "Load", um Ihren .ppk-Schlüssel zu laden
3. Wählen Sie dann im Menü Conversions > Export OpenSSH Key
4. Speichern Sie den Schlüssel z. B. als id_rsa

GitHUB

16 Verarbeitungsquittung des Versanddienstes

Eine wichtige Eigenschaft der Peppol-Übertragung ist, dass dem Sender eine rechtsverbindliche Bestätigung über die (technische) Annahme eines gesendeten Dokuments am Peppol-Zugriffspunkt des Empfängers bereitgestellt wird. Kann ein Dokument nicht erfolgreich übergeben werden, erhält der Sender stattdessen eine entsprechende Fehlermeldung.

Einführende Informationen dazu finden Sie in unserem Dokument „Peppol Basics und technische Voraussetzungen“. Dieses Dokument beschreibt die Struktur und Bedeutung der Rückmeldungen, die unser Peppol Access Point nach dem Versand eines Dokuments erzeugt.

Diese sogenannten Verarbeitungsquittungen informieren den Sender darüber, ob ein Dokument technisch erfolgreich an den Peppol Access Point des Empfängers übergeben werden konnte. Bei fehlgeschlagener Übergabe wird eine strukturierte Fehlermeldung mit technischen Details bereitgestellt. Ziel ist es, Schnittstellen-Nutzern eine verlässliche Grundlage zur automatisierten Weiterverarbeitung der Rückmeldungen zu geben, beispielsweise zur Fehlerbehandlung oder zum Monitoring.

16.1 Einfluss der verwendeten Schnittstelle

Die konkrete Form und Struktur der Antwortnachricht hängen davon ab, welche Schnittstelle für den Versand genutzt wird. Der Versanddienst liefert die Rückmeldedaten in den Formaten XML oder JSON. An der REST-API bestimmt der Client das Format zur Laufzeit über den HTTP-Header „Accept“. Bei SFTP wird das Ausgabeformat vorab konfiguriert – standardmäßig wird XML verwendet.

Folgende Unterschiede bestehen:

Schnittstelle	Typ der Antwort	Besonderheit
SFTP (asynchron)	Rückmeldung verzögert, per Datei im Zielverzeichnis	Als Format wird XML verwendet, die Quittung wird im Zielverzeichnis abgelegt
REST-API (synchron)	Antwort direkt bei Aufruf	Formatwahl per Programm über Accept-Header oder application/json oder application/xml. Der HTTP-Statuscode repräsentiert den Grundstatus

16.2 HTTP-Statuscodes und ihre Bedeutung

Die eigentlichen Inhalte der Quittung (z. B. Erfolgs- oder Fehlermeldung im Detail) stehen im Response-Body (REST) bzw. in der XML-Datei (SFTP). Im REST-Fall nutzt die HTTP-Schnittstelle die Standardstatuscodes zur Einordnung des Rückgabergebnisses, so dass Clients schon ohne Analyse des Inhalts zwischen Fehlerarten unterscheiden können:

Code	Bedeutung
200 - OK	Dokument technisch angenommen

400 – Bad Request	Fehlerhafte Eingabe (Client-Fehler, z. B. ungültiges XML, fehlende Pflichtfelder)
504 – Gateway Timeout	Antwort vom Empfänger-Zugangspunkt (C3) ausstehend (Netzwerkproblem, nicht erreichbar)

Tabelle 8: Standardstatuscodes

Im Kontext eines Peppol Access Points mit REST-API bedeutet Statuscode 200, dass die eingereichte Rechnung oder Nachricht technisch erfolgreich verarbeitet wurde und der Peppol Access Point des Absenders das Dokument korrekt an den Peppol Access Point des Empfängers weitergeleitet hat. Statuscode 200 bedeutet nicht, dass der finale Empfänger (C4) die Rechnung auch „gelesen“ oder „gebucht“ hat – es geht allein um die technische Annahme durch C3.

16.3 Datenstruktur der Verarbeitungsquittung

Die hier dargestellte Antwort-Datenstruktur tritt bei Nutzung der **asynchronen SFTP-Schnittstelle** auf. Sie enthält zusätzlich einen „**Envelope**“, einen strukturierten Umschlag mit Metadaten zur Übertragung (z. B. Zeitstempel, Dokument-IDs, Statusmeldungen).

An der **synchronen REST-API-Schnittstelle** beschränkt sich die Antwort auf den eingebetteten Kernbestandteil `PoneResponse`, ohne Umschlagstruktur.

Im **JSON-Format** entspricht die äußere geschweifte Klammer

- dem Objekt bei asynchroner Übergabe, bzw.
- dem Objekt bei synchroner Übertragung.

Das folgende Beispiel zeigt die typische Struktur einer **Verarbeitungsquittung im Erfolgsfall**, wie sie vom Peppol Access Point über die asynchrone Schnittstelle (SFTP) im XML-Format bereitgestellt wird:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TransferServiceReport>
  <OrganisationName>012345</OrganisationName>
  <DocumentContentName>m_d6f655amgei0w4di_012345.out</DocumentContentName>
  <PoolingTime>2022-04-13T08:54:40.707Z</PoolingTime>
  <PoneResponse>
    <ExternalMessageId>b16d5333-6926-466a-87f3-634419ae46b1@v1r21872.ref.erechnung.service.itzbund.net</ExternalMessageId>
    <Status>SUCCESS</Status>
    <SendTime>2022-04-13T08:56:34.865Z</SendTime>
    <DocumentNumber>123456XX</DocumentNumber>
    <DocumentUuid>4a6e8275-1065-41c1-8626-fa1f595a72b3@procurement-one.de</DocumentUuid>
    <SenderId>SI123456</SenderId>
    <SenderPeppolId>9930:DE811807949</SenderPeppolId>
    <ReceiverId>ZRE123456</ReceiverId>
    <ReceiverPeppolId>0204:991-55555PEPP0-82</ReceiverPeppolId>
    <Messages>
      <Message>
        <Code>validation</Code>
        <Severity>INFO</Severity>
        <Category>validation</Category>
        <ShortDescription>[BR-CL-24]-For Mime code in attribute use MIMEMediaType.</ShortDescription>
        <Description>[info] [BR-CL-24] in /:Invoice[1]/cac:AdditionalDocumentReference[1]/cac:Attachment[1]/cbc:EmbeddedDocumentBinaryObject[1] [BR-CL-24]-For Mime code in attribute use MIMEMediaType.</Description>
      </Message>
    </Messages>
  </PoneResponse>
</TransferServiceReport>

```

Das folgende Beispiel zeigt die Rückmeldung im **Fehlerfall**, wenn ein Dokument **aufgrund eines Validierungsfehlers** vom Peppol Access Point abgelehnt wird. Die Quittung wird über die **asynchrone Schnittstelle** im XML-Format bereitgestellt:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TransferServiceReport>
  <OrganisationName>T536</OrganisationName>
  <DocumentContentName>m_49303u9sr5h0y1un_T536.out</DocumentContentName>
  <PoolingTime>2024-01-30T13:36:53.258Z</PoolingTime>
  <PoneResponse>
    <Status>FAIL</Status>
    <DocumentNumber>RR123456</DocumentNumber>
    <DocumentUuid>a4ba94b9-8ae8-4d86-8035-dbbff9005fe2@procurement-one.de</DocumentUuid>
    <SenderId>SI123456</SenderId>
    <SenderPeppolId>9930:DE811807949</SenderPeppolId>
    <ReceiverId>BI123456</ReceiverId>
    <ReceiverPeppolId>9930:DE811807949_TEST</ReceiverPeppolId>
    <Messages>
      <Message>
        <Code>validation</Code>
        <Severity>ERROR</Severity>
        <Category>validation</Category>
        <ShortDescription>[PEPPOL-COMMON-R049] - Swedish organization number MUST be stated in the correct format. - in /:Invoice[1]/cac:AccountingSupplierParty[1]/cac:Party[1]/cbc:EndpointID[1]</ShortDescription>
        <Description>[fatal_error] [PEPPOL-COMMON-R049] in /:Invoice[1]/cac:AccountingSupplierParty[1]/cac:Party[1]/cbc:EndpointID[1] Swedish organization number MUST be stated in the correct format. Test=string-length(normalize-space()) = 10 and string(number(normalize-space())) != 'NaN'</Description>
      </Message>
    </Messages>
  </PoneResponse>
</TransferServiceReport>

```

16.4 Feldbeschreibung

Die nachfolgende Tabelle erläutert die Bedeutung und Funktion der einzelnen Felder im `TransferServiceReport`, das vom Peppol Access Point (C2) nach Verarbeitung eines Versandauftrags bereitgestellt wird. Diese strukturierte Rückmeldung dient der technischen Nachvollziehbarkeit und ermöglicht es Schnittstellen-Nutzern, die Verarbeitungsergebnisse gezielt weiterzuverarbeiten (z. B. Monitoring, Logging, Fehlerbehandlung).

Feldname	Beschreibung
OrganisationName	Bezeichnung der sendenden Organisation
DocumentContentName	Dateiname des zu sendenden Dokuments bei Übergabe an C2
PoolingTime	Zeitpunkt der Verarbeitung durch den Versanddienst – z. B. Abholzeitpunkt bei asynchronem Transfer.
PoneResponse	Antwortstatus des internen Versanddienstes (z. B. erfolgreich, abgelehnt, in Bearbeitung)
ExternalMessageld	Vom externen (hier empfangenden) Peppol Access Point (C3) vergebener, einzigartiger Identifikator für das Dokument (bzw. die Nachricht); existiert nur bei erfolgreichem Versand
DocumentNumber	Dokument-/Rechnungsnummer, aus XML-Datei entnommen (UBL-Business terms: BT-1)
DocumentUuid	Von unserem (hier sendenden) Peppol Access Point (C2) vergebener, einzigartiger Identifikator für das Dokument im Versandprozess
SenderId	Technische Kennung des Absenders im internen System – unabhängig von Peppol, aus XML-Datei entnommen (UBL-Business terms: BT-90)
SenderPeppolId	Eindeutige Peppol-ID des Absenders, inkl. Präfix (z. B. 9930:DE123456789), auf Basis der im XML-Dokument hinterlegten Endpoint-ID generiert (UBL-Business terms: BT-34)
ReceiverId	Interne Empfängererkennung, z. B. Mandanten- oder System-ID des adressierten Empfängers, aus XML-Datei entnommen (UBL-Business terms: BT-90)
ReceiverPeppolId	Eindeutige Peppol-ID des Empfängers, inkl. Präfix (z. B. 0204:900000000-03083-72), auf Basis der im XML-Dokument hinterlegten Endpoint-ID generiert (UBL-Business terms: BT-49)
Status	Gesamtstatus des Versandvorgangs an Empfänger Peppol Access Point (C3): SUCCESS oder FAIL
SendTime	Zeitstempel der tatsächlichen Übergabe des Dokuments an empfangenden Peppol Access

	Point (C3); existiert nur bei erfolgreichem Versand
Messages:	Zusätzliche Nachrichten (Messages) informieren den Sender über Hinweise, Warnungen oder Fehler, die im Rahmen der Dokumentverarbeitung aufgetreten sind.
Code	Textkonstante zur formalen Beschreibung der Meldungsbedeutung (siehe Liste der erwartbaren Fehlercodes)
Description	Ausführlicher Nachrichtentext in Klarform zur Protokollierung oder Anzeige.
Category	Einordnung des technischen Kontexts der Meldung. Zulässige Werte: Wertemenge: • <code>client_error</code> ... Kennzeichnung eines Client-Fehlers • <code>internal_error</code> ... Kennzeichnung eines Fehlers auf Seiten des Dienstbereitstellers • <code>Peppol</code> ... Nachricht stammt vom Peppol-Empfänger • <code>receiver_error</code> ... Kennzeichnung eines Fehlers durch Probleme mit dem Peppol-Empfänger (z.B. Peppol-Empfänger ist nicht erreichbar) • <code>validation</code> ... Nachricht stammt von der Dokument-Validierung
Severity	Einstufung der Kritikalität: INFO (Hinweis), WARN (Warnung), ERROR (Fehler)
ShortDescription	Optionaler Kurztext zur schnellen Einordnung der Meldung

Tabelle 9: Feldbedeutungen in der XML-Antwort-Datenstruktur des Peppol Access Points (C2)

16.5 Liste der erwartbaren Fehlercodes

Die nachfolgende Tabelle erläutert die wichtigsten Fehlercodes innerhalb von Antwortnachrichten, die der Peppol Access Point bei technischen oder fachlichen Problemen zurückgeben kann.

Nicht alle Codes treten zwangsläufig an jeder Schnittstelle oder bei jedem Fehlerfall auf. Bei Fehlern der Kategorie „Peppol“ wird der Fehlercode der empfangenden Gegenstelle verwendet.

Fehlercode Name	Bedeutung/ Hinweise zur Fehlerbehebung
bad_parameter	Eine an der API-Spezifikation festgelegte Beschränkung des Wertebereichs eines Parameters wurde verletzt. /

	Prüfen und korrigieren Sie den Wert des betroffenen Eingabeparameters.
missed_parameter	Ein erforderlicher Schnittstellen- Parameter wurde nicht übergeben. / Prüfen Sie die übergebenen Daten. (Tritt an reinen Dateischnittstellen in der Regel nicht auf.)
necessary_user_role_missed	Dem Benutzer fehlt die benötigte Benutzerrolle für den Zugriff oder eine Servicefunktion. / Kontaktieren Sie den Kundenservice und geben die <code>DocumentUuid</code> an.
network_error	Während der Dokumentverarbeitung kam es zu einem Netzwerkfehler, die weitere Verarbeitung wird dadurch verhindert. / Stellen Sie das Dokument später erneut zum Versand bereit. (Sendewiederholungen des Dienstleisters funktionieren nicht.)
no_document	Das Dokument ist nicht vorhanden oder unvollständig. / Stellen Sie sicher, dass das Dokument formal korrekt und vollständig ist.
no_document_type	Der Dokumenttyp ist nicht identifizierbar. / Prüfen Sie, ob die <code>CustomizationID</code> korrekt im Dokument enthalten ist.
no_document_process	Peppol UBL-Element <code>ProfileID</code> fehlt. / Ergänzen Sie das <code>ProfileID</code> -Element im UBL-Dokument.
no_receiver	UBL-Dokument enthält keine gültige Peppol-Empfängeradresse. / Fügen Sie eine gültige Peppol-Adresse (Peppol-ID) hinzu.
no_sender	UBL-Dokument enthält keine gültige Peppol Sender-Adresse. / Fügen Sie eine gültige Peppol-Adresse (Peppol-ID) hinzu.
no_valid_Peppol_id	Ungültige Peppol-ID (z. B. falsche Syntax). / Überprüfen Sie die Peppol-Adressen (Peppol-ID) im Dokument.
participant_not_registered	Der Empfänger ist im Peppol-Netzwerk nicht registriert. / Prüfen Sie die im Dokument angegebene Empfängeradresse.
receiver_endpoint_unavailable	Der Empfänger ist nicht erreichbar oder antwortet unverständlich. / Versenden Sie das Dokument später erneut, da Sendewiederholungen des Dienstleisters nicht zum Erfolg führen. Wenn der Fehler bleibt: Kontaktieren Sie den Kundenservice und geben Sie die <code>DocumentUuid</code> an.
receiver_not_registered	Der Empfänger ist für den angegebenen Dokumenttyp nicht registriert. / Stellen Sie sicher, dass der Empfänger den angegebenen Dokumenttyp akzeptiert.
unexpected_error	Ein interner Fehler ist aufgetreten. / Versenden Sie das Dokument später erneut. Bei wiederholtem Fehler: Kontaktieren Sie den Kundenservice unter Angabe der <code>DocumentUuid</code> .
unknown_schema_code	Ein verwendeter Schema-Code ist nicht bekannt. / Vergleichen Sie die verwendeten Codes an der Peppol-Sender-Adresse oder der Peppol-Empfänger-Adresse mit der aktuellen Liste der gültigen Participant Identifier Schemes .

unsupported_document_type	Der Dokumenttyp wird vom Dienstleister nicht unterstützt. / Prüfen Sie den <code>Document Type Identifier</code> .
unsupported_document_type_receiver	Der Empfänger unterstützt den übergebenen Dokumenttyp nicht. / Fragen Sie beim Empfänger nach den unterstützten Formaten.
unsupported_media_type	Der angeforderte Medientyp wird nicht unterstützt./ Stellen Sie sicher, dass im Client-Request <code>application/xml</code> oder ggf. <code>application/json</code> verwendet wird.
unsupported_vat_number	Die im Dokument genannte VAT-Nummer des Senders wird nicht unterstützt. / Überprüfen Sie die eingetragene Umsatzsteuer-Identifikationsnummer.
validation	Der Validator meldet Hinweise oder Fehler. / Korrigieren Sie Fehler mit dem Schweregrad <code>ERROR</code> , bevor Sie das Dokument erneut versenden. Meldungen mit <code>INFO</code> oder <code>WARN</code> erfordern keine Korrektur.
wrong_encoding	Die Zeichenkodierung des Dokuments ist ungültig. / Nutzen Sie UTF-8 als Standard-Encoding für Peppol-Dokumente.
xml_parse	Das Dokument ist kein gültiges XML. / Überarbeiten Sie die Struktur und stellen Sie sicher, dass es ein wohlstrukturiertes XML ist.
EBMS:0004	Der empfangende Peppol Access Point meldet einen Fehler mit der Kategorie „Peppol“ und der Bedeutung „other“. Im Feld <code>Description</code> finden Sie in der Regel zusätzliche, empängerspezifische Hinweise. / Warten Sie auf die Behebung durch den Empfänger und versuchen Sie den Versand später erneut. Alternativ: Kontaktieren Sie den Adressaten und informieren Sie ihn über den technischen Fehler auf seiner Seite.

Tabelle 10: Fehlercodes in der XML-Antwort-Datenstruktur des Peppol Access Points (C2)

17 Glossar

Die folgende Tabelle erläutert zentralen Begriffe und bietet weiterführende Quellenangaben.

Begriff	Erläuterung	Weiterführende Quellen
FileZilla	Lokales Programm (Client) mit Benutzeroberfläche (Graphical User Interface, GUI) zur Benutzung eines FTP-Dateiübertragungskanal	https://filezilla-project.org/ https://filezilla-project.org/versions.php (Englisch)
FTP	File Transport Protocol Standardisiertes, weltweit eingesetztes Kommando-basiertes Protokoll zur Nutzung eines FTP-Dateiübertragungskanal	https://de.wikipedia.org/wiki/SSH_File_Transfer_Protocol https://de.wikipedia.org/wiki/File_Transfer_Protocol https://en.wikipedia.org/wiki/Comparison_of_FTP_client_software (Englisch)
HTTPS	Hypertext Transfer Protocol Secure Abgesichertes Internetprotokoll, genutzt für die Kommunikation mit der REST API	https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure
Peppol	Rechtlich eigenständiger europäischer Verein zur Steuerung des Peppol-Datennetzwerkes für die Übertragung maschinenlesbarer kaufmännischer Geschäftsdokumente zwischen Unternehmen und Behörden (B2G) und zwischen Unternehmen untereinander (B2B).	https://Peppol.org/ (Englisch)
Public Key Verfahren	Verschlüsselungsverfahren mit zwei Schlüsseln, bei dem nur der Empfänger entschlüsseln kann (privater Schlüssel / private key), aber jeder Unbekannte verschlüsseln kann (öffentlicher Schlüssel / public key).	https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem https://de.wikipedia.org/wiki/Public-Key-Infrastruktur
REST	Representational State Transfer: Ein etabliertes Designmodell für Schnittstellen in der Softwarearchitektur	https://de.wikipedia.org/wiki/Representational_State_Transfer
OpenAPI	Ein etabliertes Verfahren zur Dokumentation von Schnittstellen	https://de.wikipedia.org/wiki/OpenAPI https://de.wikipedia.org/wiki/OpenAPI
SFTP	Secure-FTP: Ein um das Verschlüsselungsverfahren erweitertes FTP-Protokoll (siehe FTP).	https://de.wikipedia.org/wiki/SSH_File_Transfer_Protocol
SSH	Secure Shell: Verschlüsselungsprotokoll zur sicheren	https://de.wikipedia.org/wiki/Secure_Shell

	Kommunikation zweier Partner über ein ungesichertes Netzwerk (Internet)	https://de.wikipedia.org/wiki/Secure_Shell
TLS-Zertifikat	Transport Layer Security: Protokoll zum Austausch von Schlüsseln, sicherheitsrelevante Erweiterung des http Protokolls	https://de.wikipedia.org/wiki/Transport_Layer_Security
ViDA	VAT in the Digital Age: Mehrwertsteuer (VAT) im digitalen Zeitalter – Initiative der EU zur Förderung der Digitalisierung bei der Mehrwertsteuererhebung	https://taxation-customs.ec.europa.eu/taxation/vat/vat-digital-age-vida_en (Englisch)
Webhook	Verfahren, damit eine Schnittstelle ihren Gegenpart aktiv informieren kann, wenn eine Aktion erforderlich ist.	https://de.wikipedia.org/wiki/Webhook

Deutsche Telekom MMS GmbH

Riesaer Straße 5

01129 Dresden

Telefon: +49 3 51 - 28 20 - 0

Internet: www.telekom-mms.com