

ELSTER-Transfer

Schnittstellen von ELSTER-Transfer

ETR Version 25.07
Dokumentversion: 24

Veröffentlicht am 12.08.2025

Herausgeber: Bayerisches Landesamt für Steuern

Inhaltsverzeichnis

1 Dateiimportschnittstelle	2
1.1 Importfunktion für JSON-Adressierungsdateien	2
1.1.1 Versand von Daten zu einem Partner (Sendeaufträge)	2
1.1.2 Bereitstellen von Dokumenten in die Postfächer von Anwender	3
1.1.3 Allgemeine Anmerkungen	12
1.1.4 Fehlgeschlagener Import	12
1.1.5 Automatische Bereinigung der Ausgangsdateien nach dem Versand / nach der Bereitstellung	13
1.2 Importfunktion für Sendeaufträge anhand von Beschreibungsdateien	13
2 REST-Schnittstelle	15
2.1 Wichtige Änderungen der Schnittstelle	15
2.1.1 Änderungen mit ETR-Version 25.07	15
2.1.2 Änderungen mit ETR-Version 25.03	15
2.1.3 Änderungen mit ETR-Version 25.01	15
2.1.4 Änderungen mit ETR-Version 24.07	16
2.1.5 Änderungen mit ETR-Version 24.04	16
2.1.6 Änderungen mit ETR-Version 3.6.0	17
2.1.7 Änderungen mit ETR-Version 3.5.0	18
2.1.8 Änderungen mit ETR-Version 3.4.0	18
2.1.9 Änderungen mit ETR-Version 3.3.0	18
2.1.10 Änderungen mit ETR-Version 3.2.0	19
2.1.11 Änderungen mit ETR-Version 3.1.0	19
2.1.12 Änderungen mit ETR-Version 3.0.0	19
2.1.13 Änderungen mit ETR-Version 2.3.0	21
2.1.14 Änderungen mit ETR-Version 2.2.0	22
2.1.15 Änderungen mit ETR-Version 2.1.0	22
2.2 OpenAPI.....	22
2.3 Swagger UI.....	23
2.4 Trefferermittlung bei Datenabholung und "Rückantworten".....	23
2.5 Größenbeschränkungen für Uploads per REST-API	24
2.5.1 Größenbeschränkungen für Sendeaufträge	24
2.5.2 Größenbeschränkungen für Bereitstellungsaufträge.....	24
2.6 Statusmodelle	25
2.6.1 Statusmodell für Sendeaufträge.....	25
2.6.2 Statusmodell für Einzelabholaufträge.....	26
2.6.3 Statusmodell für Dauerabholaufträge.....	27
2.6.4 Statusmodell für Bereitstellungsaufträge.....	29
2.7 Sicherheitsbetrachtungen	33
2.7.1 Remote Zugriff	33
2.7.2 SSL-Verschlüsselung und Authentifizierung	33
2.8 Speicherplatzverwaltung	33
2.8.1 Automatische Bereinigung der Ausgangsdateien nach dem Versand / nach der Bereitstellung	33
2.9 Hilfe für die Entwicklung gegen die ETR-REST-Schnittstelle	33
2.9.1 Werkzeuge	34
2.9.2 How-To: Grundlagen	34
2.9.3 How-To: Zugriffsklassen für Client-Implementierungen generieren	34
2.9.4 How-To: Mock-Server für systemübergreifende Tests generieren.....	36
3 ELSTER-Transfer Sandbox-Variante für Entwickler.....	37
3.1 Hintergrund.....	37
3.2 Besonderheiten	37
3.2.1 Besonderheiten bei der Installation	38
3.2.2 Besonderheiten im Verhalten der Anwendung	38
3.3 Zurücksetzen einer "Sandbox"-Installation in den Initialzustand	38

1 Dateiimportschnittstelle

1.1 Importfunktion für JSON-Adressierungsdateien

Aufträge für den Versand von Dateien oder für die Bereitstellung von Dokumenten in die Postfächer von Anwender können neben der Browseroberfläche auch via json-Adressierungsdateien aus dem Importverzeichnis importiert werden. Der ETR-Dienst überprüft alle 10 Sekunden (konfigurierbar in der application.yml) das Importverzeichnis auf darin befindliche Adressierungsdateien. Wird eine Datei gefunden, dann wird versucht, diese als Sendeauftrag zu importieren.

Eine Datenübermittlung via json-Dateiimportschnittstelle besteht immer aus zwei Bestandteilen:

- ⇒ Nutzdaten
- ⇒ json-Adressierungsdatei

Die **Nutzdaten** müssen ins Ausgangsverzeichnis abgelegt werden. Dieses Verzeichnis ist in den Einstellungen der ETR-Anwendung konfigurierbar. Darunter sind Unterverzeichnisse zugelassen.

- ⇒ Beispiel Windows: C:/Users/<Nutzer>/Documents/ELSTER-Transfer/ausgang/
- ⇒ Beispiel Linux: /home/etr/elster-transfer/ausgang/

Die **json-Adressierungsdatei** muss ins Importverzeichnis abgelegt werden.

- ⇒ Beispiel Windows: C:/Users/<Nutzer>/Documents/ELSTER-Transfer/import/
- ⇒ Beispiel Linux: /home/etr/elster-transfer/import/

1.1.1 Versand von Daten zu einem Partner (Sendeaufträge)

Um Dateien zu einem Partner zu senden, müssen zwingend drei Informationen angegeben sein: nutzdatentyp, empfaenger, quelldatei.

Beispiel einer JSON-Datei für den Import eines Sendeauftrags ("test.json")

```
{
  "nutzdatentyp": "TOT",
  "empfaenger": "BY",
  "quelldatei": "nutzdaten.zip"
}
```

Namen/Werte in der json-Adressierungsdatei für Sendeaufträge

nutzdatentyp	ehem. ElsterFT-Verfahrenskürzel / Datenart z. B. TOT, CovidNovHilfeBescheid, VOBU, KOFZ, SEPA, ...
empfaenger	Kürzel des adressierten Partners z. B. BB, BE, BW, BY, HB, HE, ...
quelldatei	verweist auf die existierende, zu sendende Datei im Ausgangsverzeichnis

1.1.2 Bereitstellen von Dokumenten in die Postfächer von Anwender

Hinweis

Bitte beachten Sie, dass diese Möglichkeit nur speziell berechtigten Nutzern zur Verfügung steht.

Über die ELSTER-Transfer-Anwendung können spezielle ELSTER-Transfer-Nutzer Verwaltungsakte, die elektronisch bekannt zu geben sind, in das elektronische Postfach eines Steuerpflichtigen oder Unternehmens bereitstellen. Diese Dokumente (PDF-Format) können von den Empfängern dann eingesehen oder heruntergeladen werden. Diese Dokumente können in der aktuellen Ausbaustufe Mitteilungen und Bescheide (rechtsverbindliche Bekanntgabe nach § 122a AO) mit steuerlichem Kontext (z.B. Gewerbesteuerbescheide) oder sonstige Mitteilungen sein, bzw. Bescheide, wie z.B. eine Baugenehmigung.

Um Dokumente in das Postfach von Anwendern bereitzustellen, sind umfangreichere Angaben notwendig.

Beispiel einer JSON-Datei für den Import eines Bereitstellungsauftrags ("bescheid-test.json", ab ETR 2.3.0)

```
{
  "accountId": "1234567890",
  "datenart": "EPBescheid",
  "verfahren": "ElsterBereitstellung",
  "anhaenge": [
    {
      "quelldatei": "Mein-Bescheid.pdf",
      "mimeType": "application/pdf"
    },
    {
      "quelldatei": "Anhang1-Tabelle.csv",
      "mimeType": "text/csv"
    }
  ],
  "bescheidDatum": "2021-12-31",
  "lieferTicket": "Lieferticket 123",
  "geschaeftszeichen": "Geschäft mit Zeichen",
  "absender": "Der Sender",
  "betreff": "Diese schöne Nachricht",
  "htmlAnschreiben": "<p>Lieber Empfänger,<br>dies ist ein<br>formatiertes Anschreiben</p>",
  "emailAdresse": "test@example.com",
  "antwortlinks": [
    {
      "url" : "http://www.Elster.de/Antwortlink",
      "beschreibung" : "Mouse - Hover-Over Text",
      "titel" : "Einspruch xxx"
    }
  ]
}
```

Namen/Werte in der json-Adressierungsdatei

Je nachdem, ob es sich bei der Bereitstellung um eine Mitteilung oder einen rechtsverbindlichen Bescheid handelt, sind unterschiedliche Felder nutzbar, Pflicht oder Optional.

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
absender	Pflichtangabe	Pflichtangabe	Pflichtangabe	Pflichtangabe	z.B. "Stadt Mühldorf"	max. 128 Zeichen
accountId	Pflichtangabe	Pflichtangabe	Pflichtangabe	Pflichtangabe	gibt den ELSTER-Account des Empfängers an (Dieser wird als AccountPseudonymId im Datenkranz bereitgestellt.)	ELSTER-Account-ID (10-stellig-numerisch) oder pseudonymisierte ELSTER-Account-ID. Pro ELSTER-Zertifikat eine eigene ID, auch wenn die ELSTER Zertifikate zu einer und derselben Organisation gehören.
anhaenge ⇒ quelldatei ⇒ mimeType	Pflichtangabe	---	Pflichtangabe	Pflichtangabe	<p>Es können mehrere Anhänge in einem gemeinsamen Auftrag und jeweils unterschiedlichen Arten von Inhalten (bisher nur PDF) bereitgestellt werden.</p> <p>quelldatei: verweist auf die existierende, zu sendende Datei im Ausgangsverzeichnis</p> <ul style="list-style-type: none"> ⇒ Länge des Dateinamens (ohne Pfad und ohne Dateierweiterung): min. 1 Zeichen, max. 50 Zeichen ⇒ Länge des Dateinamens (ohne Pfad, inkl. Dateierweiterung): min. 5 Zeichen, max. 50 Zeichen 	<p>Beschränkungen bezüglich der Dateitypen und Größenvorgaben siehe → Vorgaben zu Dateitypen und Größen der bereitgestellten Anhänge sowie der Dateinamen (Länge, Sonderzeichen)</p>

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
					<p>⇒ Beschränkungen bei Steuerzeichen, Zeichen mit besonderer Bedeutung (*, ?, \, /, :, ", <, > und) und nicht-lateinischen Sonderzeichen, die beim Empfänger zu Problemen führen können (die Dateien sollten beim Erzeugen entsprechend benannt werden)</p> <p>dateiBezeichnung: Angaben zu dieser Eigenschaft in der Importdatei werden <i>ignoriert</i>. Die eindeutige fachliche Bezeichnung des Dokuments wird stattdessen automatisch aus dem Namen der Ausgangsdatei ("quelldatei") ermittelt.</p> <p>dateinameMitErweiterung: Es wird angenommen, dass der Dateiname auf Empfängerseite dem ursprünglichen Dateinamen des Absenders entsprechen soll. Daher werden Angaben zu dieser Eigenschaft in der Importdatei <i>ignoriert</i>. Der Dateiname, den der Empfänger verwenden soll, wird stattdessen immer automatisch aus dem Namen der Ausgangsdatei ("quelldatei") ermittelt.</p>	

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
					mimeType: Inhaltstype (MIME-Type) der Datei z.B. "application/pdf", siehe auch Tabelle unten.	
antwort ⇒ antwortFrist (Datum) ⇒ zuordnungskriterium (UUID) ⇒ antwortBetreffe	Optional	Optional	Optional	Optional	Bietet dem Nachrichtenempfänger die Möglichkeit auf den bereitgestellten Bescheid/Mitteilung zu antworten. Die Antwort kann wieder mit ETR abgeholt werden.	antwortFrist: ⇒ JJJJ-MM-DD (Optional) ⇒ darf nicht in der Vergangenheit liegen zuordnungskriterium: ⇒ UUID (Optional, wird von ETR generiert, falls nicht angegeben) ⇒ muss eindeutig sein antwortBetreffe: ⇒ Liste von Betreffen (Min. 1, Max 100) antwortBetreffe.betreff: ⇒ (Min. 1, Max. 255 Zeichen) ⇒ kein Betreff darf doppelt angegeben werden
antwortlinks ⇒ url ⇒ titel ⇒ beschreibung	Optional	Optional	Optional	---	Bei der Bereitstellung von Sonstigen Mitteilungen oder Sonstigen Bescheiden besteht die Möglichkeit Antwortlinks zu übergeben. Dieser Antwortlink wird dem Empfänger in Mein ELSTER oder Mein UP angezeigt und	URL: ⇒ gültige URL-Syntax (ggf. inklusive Query-Parameter, Anker usw.) ⇒ erlaubte Protokolle:

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
					<p>gibt ihm die Möglichkeit, auf Nachrichten in seinem Postfach zu reagieren. Der Antwortlink besteht aus einer URL (url), einer Beschreibung (beschreibung), welche als Tooltip angezeigt wird und einem Titel (titel), der der Darstellung des Links als Text entspricht. URL und Titel sind dabei Pflichtangabe, wenn ein Antwortlink übergeben wird.</p> <ul style="list-style-type: none"> ⇒ Derzeit kann max. 1 Antwortlink übergeben werden. ⇒ Dieses Feld darf nicht bei der Datenart Gewerbesteuerbescheid verwendet werden. 	<p>HTTP, HTTPS, FTP ⇒ max. 32.768 Zeichen</p> <p>Titel: ⇒ max. 56 Zeichen</p> <p>Beschreibung (optional): ⇒ max. 32.768 Zeichen</p>
bescheidDatum	---	---	Pflichtangabe	Pflichtangabe	Datum des Bescheids (Pflichtangabe bei Bescheiden, bei Mitteilungen nicht vorhanden)	JJJJ-MM-DD
betreff	Pflichtangabe	Pflichtangabe	Pflichtangabe	---	<p>Bezeichnung der Nachricht</p> <ul style="list-style-type: none"> ⇒ Dieses Feld ist Pflicht bei den Datenarten EPMitteilung, EPBescheid, EPKurzmitteilung. ⇒ Dieses Feld darf nicht bei der Datenart Gewerbesteuerbescheid verwendet werden. 	max. 255 Zeichen nur E-Mail-konforme Sonderzeichen (Latin-1-Zeichensatz und einige weitere)

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
datenart	Pflichtangabe	Pflichtangabe	Pflichtangabe	Pflichtangabe	<p>Angabe der Art des Dokuments:</p> <ul style="list-style-type: none"> ⇒ Übermittlung von Mitteilungen: "EPMitteilung" ⇒ Übermittlung von Kurzmitteilungen ohne Anhang: "EPKurzmitteilung" ⇒ Rechtsverbindliche Bescheide: "EPBescheid" ⇒ Rechtsverbindliche Gewerbesteuerbescheide: "Gewerbesteuerbescheid" 	"EPMitteilung", "EPKurzmitteilung", "EPBescheid", "Gewerbesteuerbescheid"
emailAdresse	---	---	Optional	Pflichtangabe	<p>Notwendig beim Versand von rechtsverbindlichen Bescheiden.</p> <p>⇒ Ausnahme bei Datenart <u>EPBescheid</u>: Wird für EPBescheid keine E-Mail Adresse angegeben, wird die E-Mail Adresse des ELSTER-Benutzerkontos verwendet.</p>	max. 255 Zeichen gültige E-Mail-Adresse (RFC 5321, 5322)
emailZusatzText	Optional	Optional	Optional	---	Ergänzender Freitext für die Benachrichtigungsemails, der zusätzlich zum Voragentext verwendet wird. Nur für bestimmte Datenarten (z.B. "EPMitteilung") und	max. 500 Zeichen nur E-Mail-konforme Sonderzeichen (Latin-1-Zeichensatz und einige weitere)

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
					Nutzergruppen (in Abhängigkeit der Anwendungskonfiguration) gedacht, sonst nicht zulässig. <i>VERALTET: Dieses Feld wird in einer zukünftigen Version in Zusammenhang mit einer Umstellung der automatisierten E-Mail-Verarbeitung ersatzlos wegfallen und sollte nicht mehr verwendet werden.</i>	
empfaengerreferenz	Optional	Optional	Optional	Optional; Pflichtangabe, wenn Mandantennummer vorgegeben ist	Anhand der Referenz kann der Empfänger den Auftrag zuordnen (z. B. Mandantennummer, Sachgebiet ...)	max. 255 Zeichen
geschaeftszeichen	Optional	Optional	Pflichtangabe	Pflichtangabe	<ul style="list-style-type: none"> ⇒ Optional bei Mitteilungen ⇒ Pflicht bei Bescheiden Für die Zuordnung des Falles zu einen Geschäftsvorgang kann das Geschäftszeichen angegeben werden.	max. 60 Zeichen nur E-Mail-konforme Sonderzeichen (Latin-1-Zeichensatz und einige weitere)
htmlAnschreiben	Optional	Pflichtangabe	Optional	---	Formatiertes Anschreiben welches dem Empfänger im Posteingang angezeigt wird. <ul style="list-style-type: none"> ⇒ ASCII- und UTF-8-Sonderzeichen müssen ggf. ersetzt (escaped) werden. ⇒ Nicht zulässig ist das Einbetten von Stylesheets und 	max. 5000 Zeichen eingeschränkte HTML-Syntax, siehe → formatiertes HTML-Anschreiben

Name	EPMitteilung	EPKurzmitteilung	EPBescheid	Gewerbesteuerbescheid		
					Skripten, HTML-Kommentare und -Verarbeitungsanweisungen sowie die Verwendung von HTML-Attributen.	
lieferTicket	Optional	Optional	Optional	Optional	<p>Eindeutige Kennung, um mit diesem (Antwort-)Dokument einen Bezug zu einem vorhergehenden Antrag herzustellen.</p> <p>Bitte in der aktuellen Umsetzung nur verwenden, wenn eine Angabe im ETID-Format sichergestellt werden kann.</p> <p>Beispiel aus dem steuerlichen Umfeld: Die abgegebene Gewerbesteuererklärung erzeugt ein eindeutiges Ticket. Der resultierende Gewerbesteuerbescheid enthält das gleiche Ticket damit der Anwender Bezug zwischen konkreter Erklärung und Bescheid herstellen kann.</p>	aktuell: ETID-Format (32 Zeichen) geplant: freier String
verfahren	Pflichtangabe	Pflichtangabe	Pflichtangabe	Pflichtangabe	konstanter Wert: "ElsterBereitstellung"	"ElsterBereitstellung"

Weist die Importdatei Angaben zu anderen Eigenschaften auf, die in der Tabelle nicht definiert sind, werden sie aus Gründen der Abwärtskompatibilität mit älteren ETR-Versionen beim Import ignoriert.

Vorgaben zu Dateitypen und Größen der bereitgestellten Anhänge

Die zulässige Anzahl und die Größe der angegebenen Dateien sind abhängig von der verwendeten ETR-Version:

ab ETR 2.0	⇒ max. 1 Anhang	⇒ max. 10 MiByte je Anhang
ab ETR 2.3	⇒ max. 5 Anhänge	⇒ max. 8 MiByte je Anhang
ab ETR 2.4	⇒ max. 50 Anhänge	⇒ max. 50 MiByte je Anhang ⇒ insgesamt max. 250 MiByte summiert über alle Anhänge
ab ETR 24.07	⇒ max. 100 Anhänge	⇒ max. 600 MiByte je Anhang ⇒ insgesamt max. 5 GiByte summiert über alle Anhänge
ab ETR 25.07	⇒ max. 100 Anhänge	⇒ max. 2 GiByte je Anhang ⇒ insgesamt max. 5 GiByte summiert über alle Anhänge

Die zulässigen MIME-Typen sind von der gewählten Datenart abhängig (weitere MIME-Typen werden bei zukünftigen Versionen ggf. ergänzt):

EPBescheid, EPMitteilung	ab ETR 2.0: ⇒ application/pdf ab ETR 2.3 zusätzlich: ⇒ text/csv ⇒ image/png ⇒ image/jpeg ab ETR 3.0 zusätzlich: ⇒ text/xml
Gewerbesteuerbescheid	ab ETR 2.0: ⇒ application/pdf

Ist der MIME-Typ nicht angegeben, wird für die Bestimmung aus der angegebenen Datei die MIME-Typ-Zuordnung des zugrundeliegenden Betriebssystems verwendet. Falls der MIME-Typ für die konkrete Datei auf diese Weise nicht feststellbar ist, wird der Import mit einer Fehlermeldung verweigert. Analog zur Weboberfläche muss der MIME-Typ dann explizit angegeben werden.

Formatiertes HTML-Anschreiben

Für einige Datenarten ist es ab ETR Version 2.3.0 möglich, neben der Übertragung der Dokumente/Anhänge optional jeweils ein formatiertes Anschreiben zu erfassen, welches dem Empfänger im Posteingang angezeigt wird. Dazu kann in der Importdatei im neu eingeführten Feld "htmlAnschreiben" ein unformatierter oder ein HTML-formatierter Text angegeben werden. Die unformatierte Länge ist auf 5000 Zeichen begrenzt.

Beispiel (verkürzt):

```
{
...
    "htmlAnschreiben": "<p>Lieber Empfänger,<br>dies ist
ein <em>formatiertes</em> Anschreiben</p>",
...
}
```

Wie in JSON üblich, müssen ASCII- und UTF-8-Sonderzeichen ggf. ersetzt (escaped) werden. Alternativ ist im Rahmen der HTML-Formatierung möglich, [HTML-Character-Entities](#) (→ [Liste gängiger Zeichen](#)) zur Eingabe von Sonderzeichen zu nutzen. Unformatierter Text wird automatisch in HTML umgewandelt, insbesondere werden ASCII-Zeilenumbrüche automatisch in HTML-Zeilenumbrüche ("
") konvertiert.

Bezüglich der HTML-Formatierung sind ausschließlich diejenigen Formatierungselemente ("Tags") des HTML-Standards erlaubt, die im Posteingang des Empfängers ordnungsgemäß dargestellt werden können (Textbeispiele):

- ⇒

- ⇒ <p>Absatz</p>
- ⇒ erster Punktzweiter Punkt
- ⇒ erster Punktzweiter Punkt
- ⇒ starke Hervorhebung (fett dargestellt)
- ⇒ Hervorhebung (kursiv dargestellt)

Nicht zulässig ist insbesondere das Einbetten von Stylesheets und Skripten, HTML-Kommentare und -Verarbeitungsanweisungen sowie die Verwendung von HTML-Attributen.

Für folgende Datenarten sind formatierte Anschreiben zulässig:

- ⇒ EPBescheid
- ⇒ EPMitteilung

Hinweis zur Kompatibilität mit älteren ETR-Versionen

Alternativ zur neuen "anhaenge"-Struktur wird die ältere Aufrufweise durch Angabe der "quelldatei" in der obersten Ebene in ETR auch weiterhin noch unterstützt. In diesem Fall muss die einzelne Datei zwingend im PDF-Format vorliegen.

1.1.3 Allgemeine Anmerkungen

Sie haben die Möglichkeit, die Datei absolut oder relativ zum Ausgangsverzeichnis anzugeben. Ggf. im Dateinamen vorhandene Sonderzeichen (z.B. "\" unter Windows) sind dabei wie im [JSON-Standard](#) üblich zu ersetzen.

Beispiele Windows:

- ⇒ "quelldatei": "C:/Users/<Nutzer>/Documents/ELSTER-Transfer/ausgang/12345.pdf"
- ⇒ "quelldatei": "12345.pdf".

Beispiele Linux:

- ⇒ "quelldatei": "/home/etr/elster-transfer/ausgang/12345.pdf"
- ⇒ "quelldatei": "12345.pdf"

Als Encoding für die json-Adressierungsdatei wird **UTF-8** erwartet.

- ⇒ Der Dateiname der json-Adressierungsdatei ist egal. Die Erweiterung muss allerdings auf .json enden.

Die Auswahl der für Sie zur Verfügung stehenden Nutzdatentypen/Datenarten hängen stets von Ihren individuell erhaltenen Berechtigungen ab.

1.1.4 Fehlgeschlagener Import

Schlägt ein Importvorgang fehl und kann daraus kein Sendeauftrag erstellt werden, dann wird der json-Adressierungsdatei der Suffix `.error` hinzugefügt.

Bitte prüfen Sie in diesem Fall den korrekten Aufbau sowie das Encoding der json-Adressierungsdatei, und stellen Sie sicher, dass die referenzierte Quelldatei im Ausgangsverzeichnis zu finden ist.

Die json-Adressierungsdateien von fehlgeschlagenen Importvorgängen mit Suffix `.error` werden standardmäßig automatisch nach 30 Tagen gelöscht.

1.1.5 Automatische Bereinigung der Ausgangsdateien nach dem Versand / nach der Bereitstellung

Sollten die Ausgangsdateien nicht später anderweitig (außerhalb von ETR) benötigt werden, empfehlen wir zur Verminderung des Speicherplatzbedarfs und aus Datenschutzgründen, die automatische Bereinigung der Ausgangsdateien nach dem erfolgreichen Versand zu aktivieren.

- ⇒ Aktivieren Sie dazu im Formular "Einstellungen" im Abschnitt "Datenübertragung einstellen" die Option "Originaldaten nach dem Übermitteln aus dem Ausgangsverzeichnis löschen (gilt für Bereitstellung und Datenübermittlung)"
- ⇒ Analog kann bei Nutzung der Vorkonfiguration der Einstellungen / des ELSTER-Zertifikats per "application.yml"-Konfigurationsdatei in dieser Datei die vorhandene Option "etr.einstellungen.dateienNachUebermittlungLoeschen" aktiviert werden

Temporär während der Verarbeitung erzeugte Dateien werden von der Anwendung regelmäßig automatisch bereinigt.

1.2 Importfunktion für Sendeaufträge anhand von Beschreibungsdateien

Diese Funktionalität wurde nur aufgrund Kompatibilität für Bestandsnutzer der Vorgängeranwendung eingebunden, damit der Umstieg auf ELSTER-Transfer mit möglichst wenig Aufwand durchgeführt werden konnte. **Es findet jedoch keine Weiterentwicklung an dieser Schnittstelle statt.** Aktuelle Neuerungen (SSL, BasicAuth, Bereitstellen von Dokumenten ...) und zukünftige Features können damit nicht mehr bedient werden. **Wir empfehlen ausdrücklich, diese Schnittstelle nicht mehr zu verwenden.**

Aufträge für den Versand von Dateien können auch aus sog. Beschreibungsdateien (Festlängenformat) aus dem Importverzeichnis **C:\Users\<Nutzer>\Documents\ELSTER-Transfer\import** (Windows), bzw. **/home/etr/elster-transfer/import/** (Linux) importiert werden.

Der ETR-Dienst überprüft alle 10 Sekunden (konfigurierbar in der application.yml) das Importverzeichnis auf darin befindliche Beschreibungsdateien. Wird eine Datei gefunden, dann wird versucht, diese als Sendeauftrag zu importieren.

Es werden alle Dateien berücksichtigt, außer Dateien, die auf `.error` enden. Schlägt der Import als Sendeauftrag fehl, dann wird der Beschreibungsdatei der Suffix `.error` hinzugefügt.

Verschiedene Anwendungen in den Verwaltungen erzeugen Beschreibungsdateien. Sie enthalten keine Zeilenumbrüche. Leerzeichen sind im Beispiel unten als Punkte (·) dargestellt.

Folgende Elemente werden von der ETR-Anwendung ausgewertet (alle anderen Elemente der Beschreibungsdatei werden ignoriert):

- ⇒ **Verfahren:** Muss immer ELSTERFT sein
- ⇒ **Datenart:** Im unten stehenden Beispiel TOT. Sie können nur Daten zu einem Verfahren versenden, für die Sie berechtigt wurden.
- ⇒ **Empfänger:** Im unten stehenden Beispiel BY. Sie können nur Daten zu einem Verfahren versenden, für die Sie berechtigt wurden.

Quelldatei: Im unten stehenden Windows-Beispiel **C:\Users\<Nutzer>\Documents\ELSTER-Transfer\ausgang\tot.txt**

Wichtig: Der Pfad muss immer mit dem Ausgangsverzeichnis **C:\Users\<Nutzer>\Documents\ELSTER-Transfer\ausgang** (Windows), bzw. **/home/etr/elster-transfer/ausgang/** (Linux) beginnen. Darunter sind Unterverzeichnisse zugelassen.

2 REST-Schnittstelle

Die REST-Schnittstelle von ELSTER-Transfer unterstützt die Bereitstellung von Dokumenten in die Postfächer von Anwendern sowie deren Statusabfrage. Seit Version 2.1.0 ist es außerdem möglich, die REST-API zum Übermitteln und Abholen von Daten zur und von der Steuerverwaltung zu nutzen.

Im Folgenden wird die Adresse der ELSTER-Transfer-Instanz mit <ETR-INSTANZ> bezeichnet. Diese muss mit Protokoll:Adresse:Port ersetzt werden (Beispiel: "<https://etr-server:8081>").

Bitte berücksichtigen Sie die Sicherheitsempfehlungen bzgl. Aktivierung von Authentifizierung und SSL-Verschlüsselung.

2.1 Wichtige Änderungen der Schnittstelle

HINWEIS: Die mit  gekennzeichneten Punkte erfordern eine neue Codegenerierung und ggf. Anpassungen am Client Code.

2.1.1 Änderungen mit ETR-Version 25.07

- ⇒ kleinere Ergänzungen/Korrekturen zu einzelnen Feldern, u.a.
 - fehlende "type"- und "minLength"-Angaben ergänzt
 - bei mehrwertigem Element "anhaenge" für Verfahren "ElsterBereitstellung" mit Datenart "EPKurzmitteilung":
Das Element war verpflichtend ("required") definiert, konnte aber in Anfragen bisher schon statt einer leeren Angabe alternativ auch weggelassen werden. Aus Konsistenzgründen wurde die Angabe "required" daher aus der Schnittstellenbeschreibung entfernt.
- ⇒ Erhöhung Größenlimitierung auf 2 GiByte je Anhang für die Datenarten "EPBescheid" und "EPMitteilung" bei Verwendung der Endpunkte "/rest/bereitstellungsauftrag/file-upload" und "POST /rest/bereitstellungsauftrag/v3". Die Größenlimitierungen für die älteren Versionen der Schnittstelle ("POST /rest/bereitstellungsauftrag" und "POST /rest/bereitstellungsauftrag/v2") bleiben aus technischen Gründen bestehen.
- ⇒ Feld "emailZusatzText" für Bereitstellungsaufträge als veraltet ("deprecated") markiert. Dieses Feld war bisher nur für bestimmte Datenarten und Nutzergruppen (in Abhängigkeit der Anwendungskonfiguration) angebbar. Es wird in einer zukünftigen Version in Zusammenhang mit einer Umstellung der automatisierten E-Mail-Verarbeitung ersatzlos wegfallen und sollte nicht mehr verwendet werden.

2.1.2 Änderungen mit ETR-Version 25.03

- ⇒ Anpassung am Feld "dateiBezeichnung", das als "fachliche" Bezeichnung jeden Anhang eines Bereitstellungsauftrags eindeutig identifiziert:
 - Zukünftig ist es möglich, auch Angaben mit Dateierweiterungen zu verwenden (z.B. "Bescheid vom 09.08.2024.pdf"), sofern sie den bestehenden Beschränkungen zu Länge und Sonderzeichen entsprechen. Auf diese Weise ist es möglich, einem Empfänger mehrere inhaltlich gleiche Kopien eines Dokuments z.B. einen Scan eines Papierdokuments in verschiedenen Dateiformaten zu übermitteln (z.B. Anlage 1 "Bescheid vom 09.08.2024.pdf" und Anlage 2 "Bescheid vom 09.08.2024.jpg").
 - Angaben ohne Dateierweiterungen (z.B. "Merkblatt zum Genehmigungsverfahren") sind wie bisher weiterhin möglich.

2.1.3 Änderungen mit ETR-Version 25.01

- ⇒ Ergänzungen/Korrekturen in der erläuternden Dokumentation ("description") zu einzelnen Feldern

- ⇒ **⚠** Die bereits als "veraltet" markierten Operationen (URLs) der ursprünglichen Schnittstelle für Bereitstellungsaufträge entfallen zukünftig. Bitte benutzen Sie als Ersatz die seit ETR 3.0 vorhandenen neueren Varianten der jeweiligen Operationen (URLs mit "/v2" oder "/v3"). Diese sind funktional gleichwertig. Konkret betrifft dies:
- bisher: "POST /rest/bereitstellungsauftrag"
→ neu: "POST /rest/bereitstellungsauftrag/v2" oder "POST /rest/bereitstellungsauftrag/v3"
 - Bei beiden neuen Varianten dieses POST-Aufrufs müssen im Detail Anpassungen am gesendeten JSON im REST-Client erfolgen.
 - Die neue "v3"-Variante ist im Gegensatz zu "v2" die modernere und sollte bevorzugt verwendet werden. Die "v3"-Variante unterstützt u.a. auch die Übertragung sehr großer Dokumente/Anhänge.
 - Bei der "v3"-Variante ist zu beachten, dass die Anhänge vorab in separaten HTTP-Requests gesendet werden müssen. Weitere Erläuterungen dazu siehe → "Änderungen mit ETR-Version 3.0.0".
 - bisher: "GET /rest/bereitstellungsauftrag/<Auftrags-ID>"
→ neu: "GET /rest/bereitstellungsauftrag/v3/<Auftrags-ID>" (Änderung der URL)
 - bisher: "POST /rest/bereitstellungsauftrag/<Auftrags-ID>/restart"
→ neu: "/rest/bereitstellungsauftrag/v3/<Auftrags-ID>/restart" (Änderung der URL)
 - Falls die Umstellung eines REST-Clients im Einzelfall nicht möglich ist, wenden Sie sich bitte mit einer Beschreibung Ihres Anwendungsfalls an den Support.
- ⇒ **⚠** Einige bisher beim Anlegen von Aufträgen erlaubte Werte für das Feld "testmerker" sind bei der Auftragsverarbeitung im ELSTER-Backend nicht zulässig. Diese unzulässigen Testmerker wurden entfernt. Es verbleiben die folgenden erlaubten Werte für das Feld "testmerker":
- Sendeaufträge: 700000001, 700000004
 - Abholaufträge: 700000001
 - Bereitstellungsaufträge: 700000001, 700000004
- ⇒ Implementierung der beiden neuen Benutzerrollen "USER_STEUERVERWALTUNG" und "USER_POSTFACH_2.0" wie im Handbuch "Fortgeschrittene Konfiguration" im Abschnitt "Client-Authentifizierung" beschrieben.
- Die Rolle "USER" sollte nach Möglichkeit auf die strengere Rolle "USER_STEUERVERWALTUNG" bzw. "USER_POSTFACH_2.0" geändert/eingeschränkt werden, sofern ein Benutzer nicht beide Funktionalitäten "Datenaustausch mit der Steuerverwaltung" (Sende- und Abholaufträge) bzw. "Postfach 2.0" (Bereitstellungsaufträge inkl. ihrer Rückantworten) benötigt.
 - Falls keine Benutzerauthentifizierung aktiviert ist (=in der Konfigurationsdatei "application.yml" verbleibt der Schlüssel "etr.sicherheit.authentifizierung" auf dem Vorgabewert "keine"), ist diese Änderung unerheblich.

2.1.4 Änderungen mit ETR-Version 24.07

- ⇒ Änderung am Verhalten bei Abholaufträgen (keine Änderung der Open API): Prüfung der Verfahrensanträge entfällt beim Anlegen eines neuen Abholauftrags. Somit können nun auch Abholaufträge für Datenarten angelegt werden, für die kein genehmigter Verfahrensantrag (mehr) vorliegt.
- ⇒ geänderte Größenbeschränkungen für das Hochladen von Anhängen über den Endpunkt "/rest/bereitstellungsauftrag/file-upload": zukünftig sind jeweils max. 600 MiByte (bisher: max. 50 MiByte) erlaubt

2.1.5 Änderungen mit ETR-Version 24.04

- ⇒ Nachdokumentation bestehender Funktionalität in OpenAPI

- Beschreibung zu Sicherheitsbeschränkungen/Authentifizierungsverfahren ergänzt (in Swagger-UI wird damit die Änderung der Authentifizierung z.B. für Testzwecke ermöglicht)
 - Beschreibung zu globalen Rückgabewerten (Status 400, 500, 503) für alle Operationen vereinheitlicht
- ⇒ Details-Anfrage zu einem bestehenden Abholauftrag (URL-Suffix: "/rest/abholauftrag/</d>") angepasst:
- Damit die ermittelte Auflistung vom Client sinnvoll verarbeitet werden kann, werden zukünftig nur noch die zuletzt ausgeführten Downloads/Abholungen zurückgeliefert (Vorgabe: max. 100.000).
 - Die Anzahl kann bei Bedarf durch den neuen Aufrufparameter "maxDownloads" (siehe Open API) weiter eingeschränkt werden.

2.1.6 Änderungen mit ETR-Version 3.6.0

- ⇒ Erhöhung Maximallänge für "Betreff" in Bereitstellungsaufträgen von 128 auf neu 255 Zeichen
- ⇒ Beschränkung am Feld "dateiBezeichnung" bei Bereitstellungsaufträgen aufgrund geänderter Schnittstellen zu nachgelagerten Systemen
 (⚠ die Clients sollten analog ETR an diese neuen und strengereren Vorgaben angepasst werden, um zukünftig Validierungsfehler zu vermeiden):
- Maximallänge bisher 255 Zeichen, neu 50 Zeichen.
 - zukünftig wird die Eingabe zusätzlich gegen einen regulären Ausdruck ("[- ~ ; - £ ¥ - ¥ \$ - \$ ^ - ® - ^ μ - μ ^ - » - ÿ € - œ Š - Š Ÿ - Ÿ Ž - Ž € - €] +") geprüft, welcher die Eingabe unzulässiger Sonderzeichen unterbindet. Dies ist für die korrekte Funktion von nachgelagerten Systemen erforderlich. Verboten sind damit insbesondere Steuerzeichen, nicht-lateinische Schriftzeichen und wenig gebräuchliche Unicode-Sonderzeichen.
 - Mindestlänge 1 und Pflichtfeldeigenschaft wird für "dateiBezeichnung" beibehalten. Die Prüfung auf Eindeutigkeit über alle Anhänge einer Bereitstellung hinweg wird ebenfalls beibehalten.
- ⇒ neues optionales Feld "dateinameMitErweiterung":
 Ein Vorschlag für den Dateiname, der beim Speichern des Anhangs / Dokumentteils im Dateisystem des Empfängers verwendet werden soll, kann bei der Bereitstellung von Dokumenten vom Absender zukünftig mitgegeben werden.
- Nach Möglichkeit soll der Originaldateiname im Dateisystem des Absenders verwendet werden, sodass die Benennung im Dateisystem auf Empfängerseite analog zur ursprünglichen Benennung auf Absenderseite ist.
 - Damit sichergestellt ist, dass der Dateiname bei allen potenziellen Empfängern gleich gehandhabt wird, muss er bestimmte technische Mindestkriterien (Längenvorgaben, Ausschluss von Verzeichnispfaden und bestimmten Sonderzeichen) erfüllen. Details siehe Open API.
 - Wird der Dateiname weggelassen, wird der Empfänger (falls z.B. zur Speicherung der erhaltenen Datei benötigt) versuchen, einen Dateinamen selbst zu generieren
- ⇒ Schnittstelle für das Herunterladen der abgeholtene Eingangsdateien (URL-Suffix: `/rest/abholauftrag/{auftragId}/download/{downloadId}`):
 Der Header "Content-Disposition" wird zukünftig die Originaldateinamen aus der Bereitstellung zurückliefern, sofern diese vom Absender mitgesendet werden. Hintergrund ist eine Erweiterung der Schnittstelle zu den Backend-Systemen, wodurch diese Information zukünftig vom Absender zusätzlich übermittelt werden kann. Der Header "Content-Disposition" kann von Clients u.a. dazu verwendet werden, beim Download passende Dateinamen zur Speicherung im lokalen Dateisystem zu generieren. Bisher wurden an dieser Stelle von ETR selbst Vorschläge für die Dateinamen generiert, die auf den vorhandenen Informationen der Datenabholung basieren.
- ⇒ Ergänzungen/Korrekturen in der erläuternden Dokumentation ("description") zu einzelnen Feldern; Ergänzung fehlender Beispielwerte

2.1.7 Änderungen mit ETR-Version 3.5.0

- ⇒ Erweiterungen und Anpassungen zur Unterstützung mehrerer "Anhänge" (=mehrere Teile eines fachlichen Dokuments z.B. ein Anschreiben mit mehreren zugehörigen Anhängen) als weitere logische Ebene innerhalb eines "Downloads". Bis auf Weiteres nur für die Abholung von "Rückantworten" relevant.
 - Neue Felder für Abfrageergebnisse bei Abholungen eingeführt: "anhaenge" (Detailinformationen zu Anhängen/Dokumentteilen in einer neuen Struktur "DownloadAnhangDetails") und "anzahlAnhaenge"
 - Neuer Query-Parameter "anhang" für die Download-Operation zum Herunterladen der Binärdaten abgeholter Dokumente, mit dem der Aufrufer bei Dokumenten mit mehreren Anhängen/Dokumentteilen über diesen iterieren kann
 - Bisherige Felder "dateiBezeichnung" und "dateiTyp" bleiben auf der Ebene von "Downloads" zunächst erhalten, werden aber abgekündigt und sollten aber nicht mehr verwendet werden. Sie werden zukünftig durch die gleichnamigen Felder auf Ebene der "Anhänge" ersetzt.
 - Bisherige Felder "beschreibungsdatei" und "nutzdatendatei" der Struktur "AntwortDetails" bleiben zunächst erhalten, werden aber abgekündigt und sollten nicht mehr verwendet werden. Beide Felder sind bei Rückantworten nur aus Kompatibilitätsgründen vorhanden. Statt "nutzdatendatei" sollte zukünftig, sofern notwendig, das neu eingeführte Feld "anhaenge" verwendet werden. Die "beschreibungsdatei" ist bei Rückantworten auch bisher stets leer.
- ⇒ Ergänzungen/Korrekturen in der erläuternden Dokumentation ("description") zu einzelnen Feldern
- ⇒ Neue zulässige Datenart "EPAntwort" beim Anlegen eines Abholauftrages. Kann verwendet werden, um Posteingänge erneut abzuholen.

2.1.8 Änderungen mit ETR-Version 3.4.0

- ⇒ Erweiterung aller Versionen (Originalversion, "v2", "v3") der Operation zum Anlegen eines neuen Bereitstellungsauftrags
 - um Vorgaben zum Senden einer Rückantwort durch den Empfänger
 - um ein neues fachliches Feld "Empfängerreferenz" für die Zuordnung der Dokumente durch den jeweiligen Empfänger
 - um das vorhandene Feld "Lieferticket" analog zur Web-GUI. Dieses Feld konnte per REST bisher zwar angegeben, aber nicht abgefragt werden (=Korrektur)
- ⇒ neue Kategorie "Eingang" mit 2 neuen REST-Operationen zur seitenweise Abfrage von eingehenden Dokumenten mit Filterung nach bestimmten vorgegebenen Kriterien
 - für abgeholte "Antworten" zu Bescheiden (u.a. mit Filtermöglichkeit bei der Bereitstellung der Originaldokuments/Bescheids angegebenen "Zuordnungskriteriums")
 - für per Einzel- oder Dauerauftrag bereits abgeholten Dokumenten anderer "ElsterFTDaten"-Datenarten (u.a. mit Filtermöglichkeit nach der jeweiligen Datenart)
- ⇒ Erweiterung der Rückgaben für die bestehende Datenabholung-Operation (im "Download-Info"-Datentyp)
 - neue Felder "Dateityp" und "DateiBezeichnung" für zukünftige Verwendung im Rahmen der "Postfach-2.0"-Datenübermittlung (für "ElsterFTDaten"-Datenarten vorerst noch leer)
 - Korrektur: das bestehende Feld "Downloadzeitpunkt" ist in seltenen Fehler-/Ausnahmefällen kein Pflichtfeld

2.1.9 Änderungen mit ETR-Version 3.3.0

- ⇒ Die neue Datenart *EPKurzmitteilung* wurde eingeführt und kann beim Anlegen eines Bereitstellungsauftrages über /rest/bereitstellungsauftrag/v2 und /rest/bereitstellungsauftrag/v3 verwendet werden.
Unterscheidungen zu der Datenart *EPMitteilung*:

- Bei dieser Datenart sind keine Anhänge zugelassen.
 - Das Feld HTML-Anschreiben ist ein Pflichtfeld.
- ⇒ neue Felder "fehlerNummer" und "fehlerText" in der Rückgabe zu Bereitstellungsaufträgen (nur im Zustand "DATEN_FEHLERHAFT" werden Informationen zur Fehlerursache geliefert, sofern bekannt)
- ⇒ neues Feld "emailZusatzText" (nur datenart- und konfigurationsabhängig befüllbar)
- ⇒ Ergänzungen/Korrekturen in der erläuternden Dokumentation ("description") zu einzelnen Feldern

2.1.10 Änderungen mit ETR-Version 3.2.0

Es sind keine Schnittstellenänderungen erfolgt.

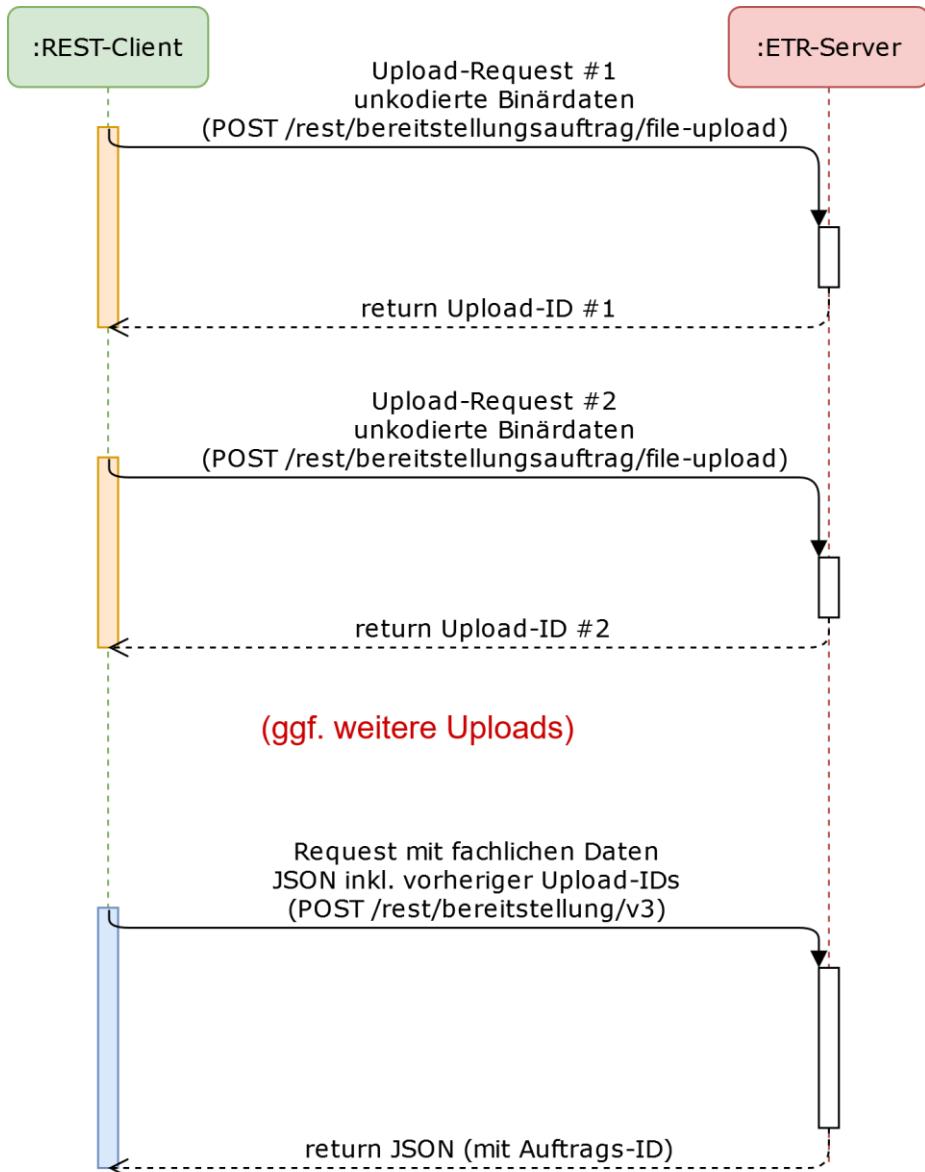
2.1.11 Änderungen mit ETR-Version 3.1.0

Es sind keine Schnittstellenänderungen erfolgt.

2.1.12 Änderungen mit ETR-Version 3.0.0

Ab ETR-Version 3.0.0 ist die Schnittstellenversion die gleiche Versionsnummer wie das ELSTER-Transfer

- ⇒ Unterstützung von Amtlichen Gemeindeschlüsseln (AGS) zur Identifikation der Gemeinde, die als Empfänger bestimmt ist, bei der Datenabholung
 - Erweiterung Operation "getAbholAuftragByld" für Datenabholung um ein neues Attribut "ags".
 - nur für relevante Datenarten wie GMBX, GMBAX, GMBVX... werden in diesem Feld zukünftig die üblichen 8-stelligen Werte (ggf. mit führender Null) für die AGS geliefert
 - Werte werden nur für zukünftig ausgeführte Datenabholungen geliefert (nicht rückwirkend für bereits durchgeführte Datenabholungen)
 - ! Änderung der zurückgelieferten Werte am HTTP-Rückgabe-Header "content-disposition" der "download"-Operation zum Herunterladen der Dokumente: wenn der AGS für das konkrete Dokument gesetzt ist, wird dieser zusätzlich als Infix in den generierten Dateinamen aufgenommen, sodass sich in diesen Fällen im HTTP-Header "content-disposition" andere Werte ergeben
- ⇒ neue Variante (REST-Operation mit Suffix "/v3") zum Anlegen von Bereitstellungsaufträgen, um zukünftig eine höhere Anzahl größerer Binärdaten/-Anhänge zu ermöglichen
 - die Übertragung der Binärdaten/Anhänge erfolgt in separaten HTTP-Requests (1 "Upload"-Request je Anhang) vor der Übertragung der fachlichen Daten (JSON). Sind n Anhänge zu übertragen, müssen also $n + 1$ HTTP-Requests ausgeführt werden.



- dabei erfolgt keine BASE64-Kodierung mehr, stattdessen werden die Binärdaten mit passender Angabe im "Content-Type"-Header z.B. "application/pdf" direkt im Request Body übertragen
- zurückgegeben wird ein JSON mit einer eindeutigen ID ("uploadId"), welche im Folge-Request mit den fachlichen Daten (JSON) im gleichnamigen Feld "uploadId" referenziert wird. Erst dort erfolgt die fachliche Definition der bereitzustellenden Daten u.a. mit Angabe der fachlichen Bezeichnungen und der Reihenfolge, wie die Binärdaten/Anhänge beim Empfänger angezeigt werden sollen.
- die separat übertragenen Binärdaten/Anhänge werden in ETR eine begrenzte Zeit (standardmäßig: 1 Tag) zwischengespeichert. Erst durch das Übertragung der fachlichen Daten (JSON) gilt die Übertragung insgesamt als abgeschlossen. Sollte diese ausbleiben, erfolgt nach Ablauf der begrenzten Speicherzeit das automatische Löschen der ggf. zuvor übertragenen Binärdaten/Anhänge.
- Erhöhung der zulässigen Anzahl Anhänge auf max. 50 Anhänge bei gleichzeitiger Beschränkung
 - Größe jedes Anhangs auf max. 50 MiByte und
 - summierte Größe der Binärdaten aller Anhänge (bei Bereitstellung von EPMitteilung und EPBescheid) auf max. 250 MiByte

- die bestehenden Aufrufvarianten (REST-Operationen) für die Bereitstellung von Dokumenten bleiben bis auf Weiteres unverändert erhalten, auch mit den geltenden geringeren Größenlimitierungen
- ⇒ Kennzeichnung Feld "testmerker" beim Anlegen von Abhol-/Sende-/Bereitstellungsaufträgen, dass diese nur der internen Verwendung dienen und zukünftig produktiv über REST nicht mehr für neue Aufträge angegeben werden sollen ([ETR-720](#), inkl. zugehörige Validierung)
- ⇒ Änderung Versionsnummernschema:
zukünftig entspricht die Versionsnummer der REST-Schnittstelle (in der OpenAPI angegeben) immer der Versionsnummer der ELSTER-Transfer-Software
- ⇒ Mindestversionsprüfung: alle REST-Operationen können zukünftig Status 503 liefern, wenn die installierte Softwareversion "veraltet" ist (Prüfung analog zur Web-GUI)
- ⇒ ⚠ geänderte Validierungsregel für Gewerbesteuerbescheide: E-Mail-Adresse für die Benachrichtigung des Empfängers zukünftig als Pflichtfeld ([ETR-708](#), analog zur Web-GUI)
- ⇒ für die Bereitstellung der Datenarten "EPBescheid" und "EPMitteilung" ist zukünftig auch das XÖV-/XML-Dateiformat zulässig (MIME-Typ "text/xml", [ETR-567](#))
- ⇒ kleinere Korrekturen und Ergänzungen zu Formulierungen
- ⇒ (diese Liste wird ggf. noch ergänzt)

2.1.13 Änderungen mit ETR-Version 2.3.0

Änderungen der ETR-Version 2.3 (Schnittstellenversion 2.1) im Vergleich mit ETR 2.2 (Schnittstellenversion 2.0):

- ⇒ ⚠ geänderte Limitierungen für die Größe der für die "ElsterBereitstellung" übertragenen Datei-Anhänge
 - max. 5 Anhänge (bisher 1 Anhang)
 - max. 8 MiByte Größe (bisher 10 MiByte)
- ⇒ optionales Mitsenden von "Anschreiben" zur Bereitstellung für die Datenarten "EPBescheid" und "EPMitteilung"
 - max. 5000 Zeichen Länge
 - optionale Verwendung folgender HTML-Formatierungselemente als Alternative zu unformatiertem Text
 - manuelle Zeilenumbrüche:

 - Absätze: <p>
 - Aufzählungslisten: mit
 - nummerierte Liste: mit
 - Hervorhebungen: und
 - keine zusätzlichen HTML-Attribute, Stylesheet-Angaben, HTML-Kommentare o.ä.
 - ASCII-Zeilenumbrüche werden, sofern vorhanden, automatisch in HTML-Zeilenumbrüche
 konvertiert
- ⇒ Korrektur der OpenAPI: Dateibezeichnung und Dateinhalt bei Anhängen sind nun auch im Schema als Pflicht definiert
(beide waren bereits vorher durch entsprechende Validierungen effektiv bereits Pflichtangaben)
- ⇒ Dokumentation: ergänzende Beschreibung zum Typ "AuftragStatus"
- ⇒ Neue Ausprägung für den Inhaltstyp (Typ "AnhangMimeType") zur Unterstützung für XML-Dateien
 - (⚠ ETR-seitig vorerst für alle Datenarten deaktiviert - zur Weiterentwicklung für XÖV in ETR 2.4 vorgesehen)

Die mit ⚠ gekennzeichneten Punkte erfordern eine neue Codegenerierung und ggf. Anpassungen am Client Code.

2.1.14 Änderungen mit ETR-Version 2.2.0

Änderungen der REST Schnittstelle Version 2.0 (ETR 2.2):

- ⇒ neue URL mit Suffix "/v2" für Bereitstellungsanfragen
(die bisherigen URLs bleiben mit "deprecated"-Markierung versehen vorerst weiterhin funktionsfähig)
 - neue strukturierte Auflistung "anhaenge"; die bestehenden Felder "dateInhalt" und "dateiBezeichnung" werden dorthin verschoben
 - Vorbereitende Änderungen zur Bereitstellung mehrere Dateien (Anhänge) im selben Auftrag, beabsichtigte Aktivierung erst mit ETR 2.3
- ⇒ Erweiterung zur Bereitstellung mehrerer (bis zu 10) Antwortlinks
- ⇒ Ergänzungen/Korrekturen in der erläuternden Dokumentation ("description") zu einzelnen Feldern

2.1.15 Änderungen mit ETR-Version 2.1.0

Änderungen der REST Schnittstelle ab ETR 2.1:

- ⇒ Titel zukünftig „ELSTER Transfer (ETR) REST-Schnittstelle“ (zusammengefasst mit ehemals „Beschreibung“)
- ⇒ Erweiterung um „Antwortlinks“ beim Anlegen eines Bereitstellungsauftrags
- ⇒ Erweiterung: neue Operation unter URL-Suffix „/rest/bereitstellungsauftrag/{id}/restart“ zum Neustarten der Übertragung von Bereitstellungsaufträgen
- ⇒ Beschreibung des Testmerkers und des Auftragsstatus nicht mehr in-line sondern als eigene Komponente (#/components/schemas/Testmarker“ bzw. „#/components/schemas/BereitstellungAuftragStatus“) in der SSb, keine Änderung am Verhalten !
- ⇒ Korrektur bei Komponente „#/components/schemas/AuftragAngelegt“, Attribut "id" ist jetzt required, ist eher eine Formsache, weil wir das Attribut bisher immer schon gesetzt hatten
- ⇒ Statt der bisherigen OperationID "create" wurden neue OperationIDs
 - createBereitstellungAuftrag
 - createAbholAuftrag
 - createSendeAuftrag

eingeführt. Dies hat zur Folge, dass generierter Code (Client oder ServerMock) neu generiert werden muss.

In dem generierten Code wird die bisherige Methode "create" dann neu "createBereitstellungAuftrag" heißen.

Verwender dieser Methode müssen entsprechend angepasst werden !

- ⇒ An einigen Stellen wurden in der Beschreibung unvollständige Beispiele und Kommentare ergänzt. Kleinere Formulierungs-/Schreibfehler wurden korrigiert.
- ⇒ neue REST-Operationen für Sendeaufträge
- ⇒ neue REST-Operationen für Abholaufträge

2.2 OpenAPI

Die Dokumentation der REST-Schnittstelle existiert als [OpenAPI](#)-Beschreibung wahlweise im JSON- und YAML-Format. Um die OpenAPI-Beschreibung von der Anwendung generieren zu lassen, kann man in der application.yml folgende Einstellung aktivieren:

```
springdoc:
  api-docs:
    enabled: true
```

Nach einem Neustart von ELSTER-Transfer kann die Beschreibung der REST-Schnittstelle unter <ETR-INSTANZ>/v3/api-docs im JSON-Format und unter <ETR-INSTANZ>/v3/api-docs.yaml im YAML-Format angezeigt werden.

2.3 Swagger UI

Swagger UI ist ein Tool zum Darstellen von OpenAPI-Beschreibungsdateien und ist Teil der ELSTER-Transfer-Anwendung. Diese kann in der application.yaml wie folgt aktiviert werden:

```
springdoc:
  api-docs:
    enabled: true
  swagger-ui:
    enabled: true
```

Nach einem Neustart von ELSTER-Transfer kann die Swagger UI unter <ETR-INSTANZ>/swagger-ui.html angezeigt werden. Standardmäßig wird direkt die Beschreibung der ELSTER-Transfer REST-Schnittstelle angezeigt.

2.4 Trefferermittlung bei Datenabholung und "Rückantworten"

Um Herauszufinden, welche abgeholten Dokumente heruntergeladen werden können, stehen die Operationen der "Eingang"-Schnittstelle zur Verfügung:

- ⇒ URL-Suffix "/rest/eingang/downloads" für abgeholte Dokumente von der Steuerverwaltung
- ⇒ URL-Suffix "/rest/eingang/antworten" für Rückantworten zu zuvor bereitgestellten Dokumenten (Postfach-2.0-Posteingang)

Dabei wird jeweils eine Suche anhand der übergebenen Query-Parameter durchgeführt und das Ergebnis als seitenweise Auflistung zurückgeliefert.

- ⇒ Die einzelnen Kriterien, die angegeben werden können, sind *optional* und *und-verknüpft*. Sie können durch den Aufrufer flexibel für den jeweils passenden Einsatzzweck zusammengestellt werden. In der Praxis erfolgt die Eingrenzung üblicherweise entweder per Datum (von/bis) oder über das bei der Bereitstellung des Originaldokuments für die Rückantwort vergebene "Zuordnungskriterium".
- ⇒ Nicht angegebene Kriterien werden bei der Abfrage nicht berücksichtigt.
→ Da in den Beispieldaten der optional zuschaltbaren Swagger-UI (s.o.) standardmäßig alle Suchkriterien mit beispielhaften Daten vordefiniert sind, müssen die Angaben für die nicht benötigten Kriterien dort also gelöscht werden.
- ⇒ Die Anzahl der zurückgelieferten Datensätze ist auf "seitenGroesse" viele beschränkt (Standardwert: 1000 Einträge je Seite). Durch Iterieren des "seitenIndex"-Parameters kann der Aufrufer ggf. weitere Ergebnisdatensätze abfragen.
- ⇒ Technisch zulässig ist es, überhaupt keine Suchkriterien oder Angaben zu "seitenGroesse"/"seitenIndex" zu verwenden. Dabei sollte aber berücksichtigt werden, dass tendenziell viele Ergebnisse ermittelt werden und die Abfrage sehr lange dauern kann.

Entspricht keiner der Datensätze den angegebenen Suchkriterien (auch wenn z.B. die Suchkriterien falsch angegeben wurden), wird der HTTP-Status 404 zurückgegeben. So kann der Aufrufer bereits beim Auslesen der HTTP-Header entscheiden, ob eine weitere Verarbeitung des Ergebnisses erfolgen soll oder nicht.

2.5 Größenbeschränkungen für Uploads per REST-API

2.5.1 Größenbeschränkungen für Sendeaufträge

Das über die REST-Schnittstelle per "Sendeauftrag" zu versendende Dokument darf max. 10 MiByte groß sein. (Über die Weboberfläche und über den Import einer JSON-Adressierungsdatei können, abhängig von ihrer Komprimierbarkeit, ggf. größere Dateien versendet werden.)

2.5.2 Größenbeschränkungen für Bereitstellungsaufträge

Bei Nutzung der REST-Schnittstelle für die Bereitstellung von Dokumenten an einen Empfänger gelten in ETR mehrere Größenvorgaben, die sich gegenseitig überlagern.

Technische Größenbeschränkungen

Bereitstellung API Version	max. Anzahl Anhänge	max. Dateigröße pro Anhang	max. Größe JSON	Dateiformat (MIME-Typ)
ursprüngliche Version	1	8 MiByte	10 MiByte	optional, Vorgabe ist "application/pdf"
V2	mehrere, per YAML steuerbar	8 MiByte	10 MiByte	optional, Vorgabe ist "application/pdf"
V3	mehrere, per YAML steuerbar	2 GiByte	(Anhänge werden über separate HTTP-Anfragen übertragen)	Pflichtangabe

Fachliche Größenbeschränkungen (auf Ebene der gewählten Datenart)

Datenart	max. Anzahl Anhänge	max. Größe aller Anhänge
EPMitteilung	100	5 GiByte
EPKurzmitteilung	0	0
EPBescheid	100	5 GiByte
Gewerbesteuerbescheid	50	40 MiByte

Konfigurative Größenbeschränkungen

YAML-Konfiguration im Auslieferungszustand	max. Anzahl Anhänge	max. Dateigröße aller Anhänge (in Summe)
	100	5 GiByte

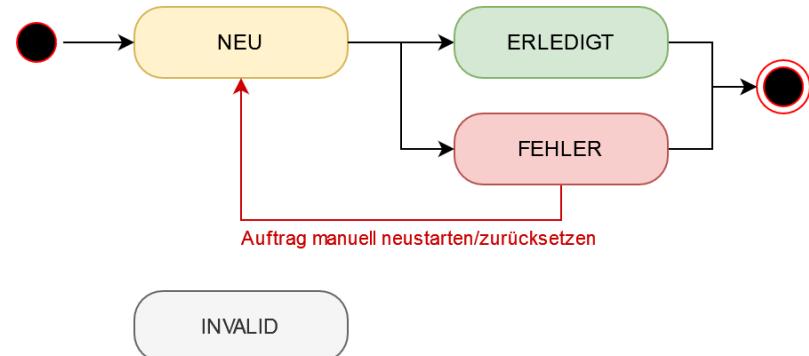
2.6 Statusmodelle

2.6.1 Statusmodell für Sendeaufträge

Statuscode (REST API)	Anzeige auf der Weboberfläche	Endzustand?
NEU Der Auftrag wurde in ETR erzeugt, aber noch nicht an den ELSTER-Eingangsserver verschickt.	"neu" oder "in Bearbeitung"	Nein
ERLEDIGT Eingangsserver haben den Datensatz angenommen. (Über die weitere Verarbeitung erfolgt bei Sendeaufträgen keine Rückmeldung an ETR!)	✓ Gesendet	Ja
FEHLER Eingangsserver haben den Datensatz aufgrund eines Fehlers <i>nicht</i> angenommen.	! Fehler	Ja, aber Neustart möglich
INVALID <i>(nur noch zwecks Kompatibilität vorhanden und für neue Aufträge nicht mehr verwendet)</i>	unbekannt	Ja

Initialer Zustand für neu angelegte Sendeaufträge ist "NEU". Dieser Zustand wird beibehalten, solange der Auftrag in ETR bearbeitet wird.

Aus Kompatibilitätsgründen erfolgt bei Zugriff per REST API (im Gegensatz zur Weboberfläche) derzeit keine separate Unterscheidung eines Zustands "IN_BEARBEITUNG".

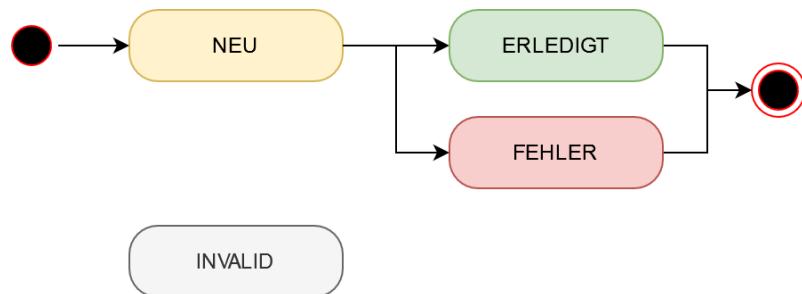


2.6.2 Statusmodell für Einzelabholaufträge

Statuscode (REST API)	Anzeige auf der Weboberfläche	Endzustand?
NEU Der Auftrag wurde in ETR erzeugt, aber die Abholung wurde noch nicht ausgeführt. <i>bis ETR 24.07.1 auch für Aufträge "in Bearbeitung" verwendet (keine Unterscheidung)</i>	"neu" oder "in Bearbeitung"	Nein
ERLEDIGT Die Abholung wurde erfolgreich ausgeführt: Alle verfügbaren Dokumente wurden vollständig heruntergeladen und der Empfang der Dokumente wurde bestätigt.	Erledigt	Ja
FEHLER Die Abholung konnte aufgrund eines Fehlers nicht vollständig ausgeführt werden.	Fehler	Ja (Neustart nicht möglich)
INVALID <i>(nur noch zwecks Kompatibilität vorhanden und für neue Aufträge nicht mehr verwendet)</i>	unbekannt	Ja

Initialer Zustand für neu angelegte Einzelabholaufträge ist "NEU". Dieser Zustand wird beibehalten, solange der Auftrag in ETR bearbeitet wird.

Aus Kompatibilitätsgründen erfolgt bei Zugriff per REST API (im Gegensatz zur Weboberfläche) derzeit keine separate Unterscheidung eines Zustands "IN_BEARBEITUNG".



Im Unterschied zu den "Sendeaufträgen" ist das Zurücksetzen im Status "Fehler" nicht möglich. Um bei Fehlern Daten erneut abzuholen, kann stattdessen ein weiterer Einzelabholauftrag für die gleiche Datenart angelegt werden.

2.6.3 Statusmodell für Dauerabholaufträge

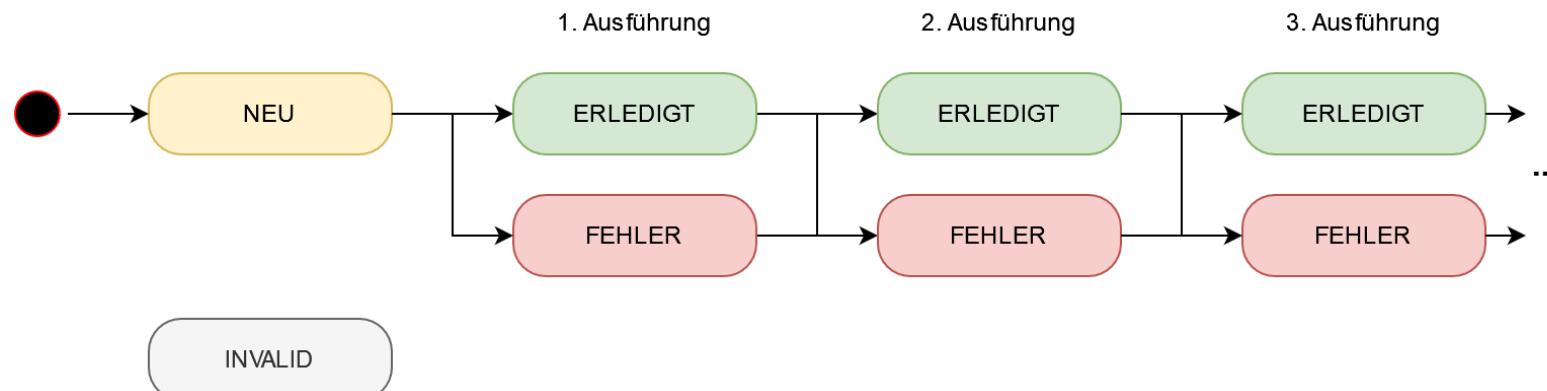
Statuscode	Anzeige auf der Weboberfläche	Endzustand?
NEU Der Auftrag wurde in ETR erzeugt, aber die erste Abholung wurde noch nicht ausgeführt.	-	Nein
ERLEDIGT Die zuletzt ausgeführte Abholung war erfolgreich: Alle verfügbaren Dokumente wurden vollständig heruntergeladen und der Empfang der Dokumente wurde bestätigt.	-	Nein
FEHLER Die letzte Abholung konnte aufgrund eines Fehlers nicht vollständig ausgeführt werden.	-	Nein
INVALID <i>(nur noch zwecks Kompatibilität vorhanden und für neue Aufträge nicht mehr verwendet)</i>	-	Nein

Initialer Zustand für neu angelegte Dauerabholaufträge ist "NEU". Dieser Zustand wird beibehalten, bis der Auftrag in ETR zum ersten Mal bearbeitet wird.

Aus Kompatibilitätsgründen erfolgt bei Zugriff per REST API (im Gegensatz zur Weboberfläche) derzeit keine separate Unterscheidung eines Zustands "IN_BEARBEITUNG".

Da ein Dauerabholaufträge wiederholt bearbeitet wird, gibt es keinen besonderen Endzustand. Die Möglichkeit des manuellen Zurücksetzens im Status "Fehler" (analog zu den "Sendeaufträgen") entfällt ebenfalls. Daueraufträge werden stattdessen so lange im eingestellten Intervall wiederholt ausgeführt, bis sie gelöscht (oder bei "Systemaufträgen" deaktiviert) werden.

Anmerkungen: Der Status eines Dauerabholauftrags ist nicht auf der Weboberfläche einsehbar. Daueraufträge können nur über die Weboberfläche angelegt werden.



2.6.4 Statusmodell für Bereitstellungsaufträge

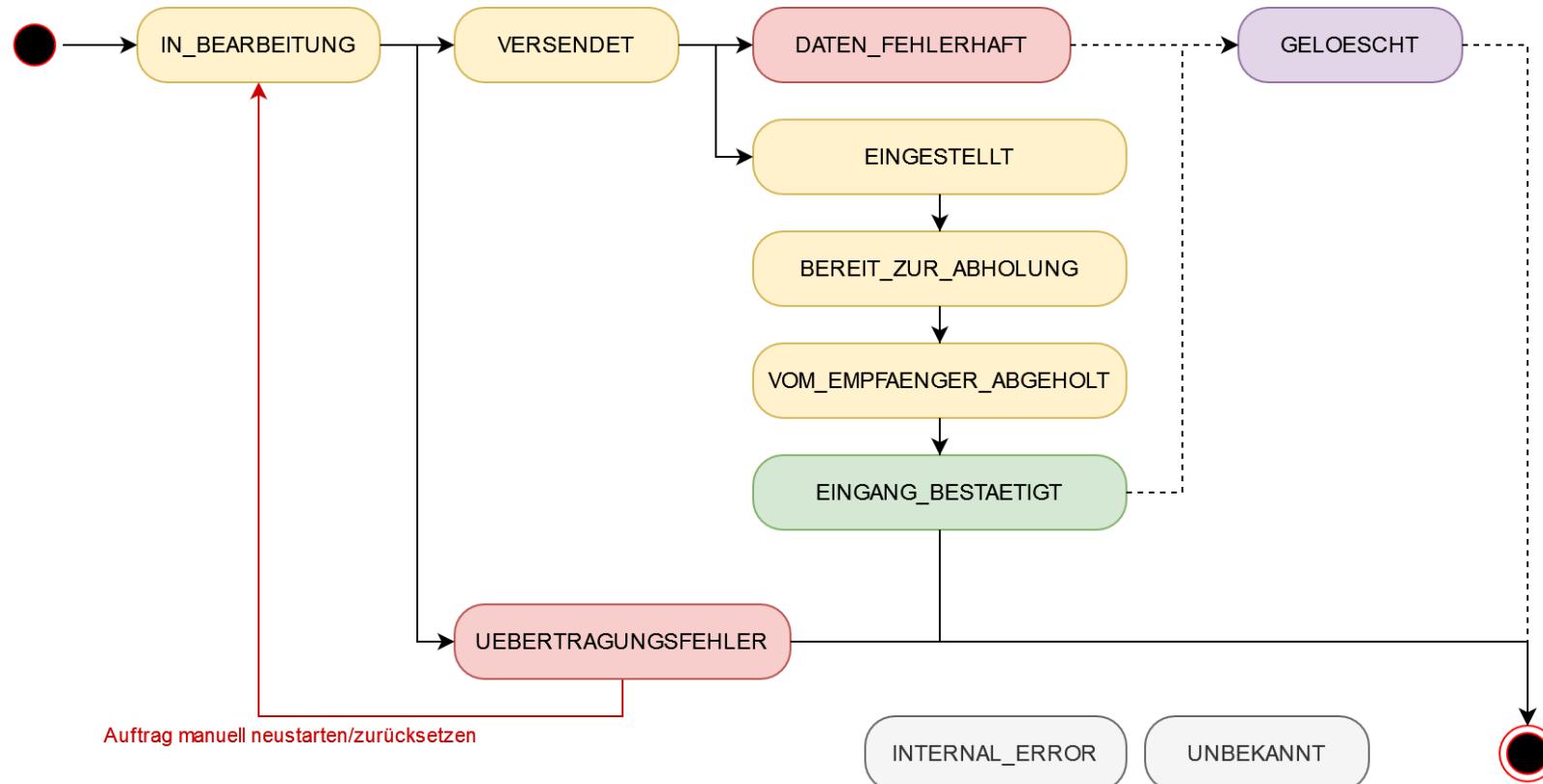
Statuscode	Endzustand?	Fehlerstatus	Feld „erstellZeitpunkt“ gefüllt?	Feld „emailVersandZeitpunkt“ gefüllt?	Feld „abrufZeitpunkt“ gefüllt?	Felder „fehlerNummer“ und „fehlerText“ gefüllt?
IN_BEARBEITUNG Auftrag in ETR erzeugt, aber noch nicht an ELSTER-Eingangsserver verschickt	Nein	Nein	Ja	Nein	Nein	Nein
VERSENDET Eingangsserver haben den Datensatz angenommen (sollten Sie kaum/nie sehen, da Statuswechsel bei ELSTER i.d.R. schneller erfolgt als die periodische Statusabfrage)	Nein	Nein	Ja	Nein	Nein	Nein
EINGESTELLT Datensatz wurde bei ELSTER in der Datenbank gespeichert, aber Bekanntgabe ist noch nicht erfolgt (sollten Sie kaum/nie sehen, da Statuswechsel bei ELSTER i.d.R. schneller erfolgt als die periodische Statusabfrage)	Nein	Nein	Ja	Nein	Nein	Nein
BEREIT_ZUR_ABHOLUNG Bekanntgabe an den Empfänger ist erfolgt. Wichtiger Status	Nein	Nein	Ja	Ja	Nein	Nein
VOM_EMPFAENGER_ABGEHOLT Datensatz wurde vom Empfänger abgeholt, aber	Nein	Nein	Ja	Ja	Nein	Nein

Statuscode	Endzustand?	Fehlerstatus	Feld „erstellZeitpunkt“ gefüllt?	Feld „emailVersandZeitpunkt“ gefüllt?	Feld „abrufZeitpunkt“ gefüllt?	Felder „fehlerNummer“ und „fehlerText“ gefüllt?
die Abholung noch nicht bestätigt. (sollte kaum/nie sehen sein, da Statuswechsel bei ELSTER i.d.R. schneller erfolgt als die periodische Statusabfrage)						
EINGANG_BESTAETIGT Empfänger hat Daten abgeholt. Abholung wurde bestätigt. Wichtiger Status	Ja	Nein	Ja	Ja	Ja	Nein
GELOESCHT Der Zustand "GELOESCHT" wird in ETR i.d.R. nicht durchlaufen.	Ja	Nein	Ja	Ja, sofern nicht zuvor "DATEN_FEHLERHAFT"	Ja, sofern zuvor "EINGANG_BESTAETIGT"	Nein
UEBERTRAGUNGSFEHLER	Ja, aber Neustart möglich	Ja	Nein	Nein	Nein	Nein
DATEN_FEHLERHAFT Z.B. Datei fehlerhaft, Virenprüfung angeschlagen, ungültiger Account oder andere auf den konkreten Datensatz bezogene Fehler. Ab ETR 3.3 werden ggf. weitere Details zur Fehlerursache in den Feldern „fehlerNummer“ und „fehlerText“ bereitgestellt.	Ja	Ja	Ja	Nein	Nein	Ja, sofern verfügbar
INTERNAL_ERROR (nur noch zwecks Kompatibilität vorhanden)	Ja	Ja	Nein	Nein	Nein	Nein

Statuscode <i>und für neue Aufträge nicht mehr verwendet)</i>	Endzustand?	Fehlerstatus	Feld „erstellZeitpunkt“ gefüllt?	Feld „emailVersandZeitpunkt“ gefüllt?	Feld „abrufZeitpunkt“ gefüllt?	Felder „fehlerNummer“ und „fehlerText“ gefüllt?
UNBEKANNT	Nein	(kann auch vorkommen, wenn die periodische Abfrage zufällig zwischen "VERSENDET"/"EINGESTELLT" passiert)	(nicht definiert; vorhandene Werte stammen i.d.R. aus dem jeweils vorherigen Status)			

Initialer Zustand für neu angelegte Bereitstellungsaufträge ist "IN_BEARBEITUNG". Dieser entspricht dem Zustand "NEU" bei Sendaufträgen. Aus Kompatibilitätsgründen erfolgt bei Zugriff per REST API (im Gegensatz zur Weboberfläche) derzeit keine separate Unterscheidung eines Zustands "NEU" für Bereitstellungsaufträge.

Wechsel vom bzw. in den Zustand "UNBEKANNT" sind immer möglich (in der folgenden Grafik nicht separat angegeben).



2.7 Sicherheitsbetrachtungen

2.7.1 Remote Zugriff

Im Auslieferungszustand nimmt die ETR-Anwendung nur Anfragen vom lokalen Rechner entgegen. Sofern der REST-Client auf einem anderen Rechner als die ETR-Anwendung laufen soll, ist es erforderlich den Remote-Zugriff zu aktivieren. Damit der Zugriff von einem entfernten Rechner möglich ist, muss ein Adressmuster in der Einstellung 'allowFrom' definiert werden, welches die zulässigen Adressen beschreibt.

Um bspw. den Zugriff von allen Adressen zu ermöglichen, muss folgende Einstellung in der `application.yml` angepasst bzw. hinzugefügt werden:

```
etr:
  access:
    allowFrom: '.*'
```

Weitere Details und Beispiele sind → im Kapitel "Remote-Zugriff aktivieren und konfigurieren" des Handbuchs "Fortgeschrittene Konfiguration" beschrieben.

2.7.2 SSL-Verschlüsselung und Authentifizierung

Die Vorgaben im Handbuch "Fortgeschrittene Konfiguration" gelten grundsätzlich auch für den Zugriff über die REST-Schnittstelle, unter anderem:

- ⇒ "TLS-/SSL-Verschlüsselung"
- ⇒ "Client-Authentifizierung"
- ⇒ "Gruppen-Modus"

Der REST-Client muss die korrekten Basic-Auth-Credentials bei jeder Anfrage mitsenden.

2.8 Speicherplatzverwaltung

2.8.1 Automatische Bereinigung der Ausgangsdateien nach dem Versand / nach der Bereitstellung

Bei ausschließlicher Nutzung der REST-API empfehlen wir zur Verminderung des Speicherplatzbedarfs und aus Datenschutzgründen, die automatische Bereinigung der Ausgangsdateien nach dem erfolgreichen Versand zu aktivieren.

- ⇒ Aktivieren Sie dazu im Formular "Einstellungen" im Abschnitt "Datenübertragung einstellen" die Option "Originaldaten nach dem Übermitteln aus dem Ausgangsverzeichnis löschen (gilt für Bereitstellung und Datenübermittlung)"
- ⇒ Analog kann bei Nutzung der Vorkonfiguration der Einstellungen / des ELSTER-Zertifikats per "application.yml"-Konfigurationsdatei in dieser Datei die vorhandene Option "`etr.einstellungen.dateienNachUebermittlungLoeschen`" aktiviert werden

Temporär während der Verarbeitung erzeugte Dateien werden von der Anwendung regelmäßig automatisch bereinigt.

2.9 Hilfe für die Entwicklung gegen die ETR-REST-Schnittstelle

Aus den bereitgestellten OpenAPI-Spezifikationen (im JSON- oder YAML-Format) lässt sich mit Hilfe von Werkzeugen für verschiedene Programmiersprachen automatisiert neben **Client-Implementierungen** zum Aufruf der REST-API u.a. für systemübergreifende Tests ein

vollständiger kleiner Web-Server generieren, der als Attrappe (engl. "**Mock**"-Server) reproduzierbar vordefinierte Testdaten zurückliefern kann.

Im Folgenden sind beispielhaft die Vorgehensweisen zur Code-Generierung von Client-Zugriffsklassen und Server-Mocks mit Hilfe des *OpenAPI Generators* über die Kommandozeile beschrieben.

2.9.1 Werkzeuge

Für den Zugriff auf die ETR-REST-API wird die Generierung von Programmcode z.B. mit Hilfe des *OpenAPI Generators* (<https://openapi-generator.tech/>) empfohlen.

- ⇒ Zur Ausführung wird eine Java-Runtime (derzeit >= Java 8; Stand: Juli 2021) benötigt.
- ⇒ Die [Installation](#) des Generators ist bspw. über *npm* oder durch das Herunterladen der JAR-Datei aus [einem öffentlichen Maven-Repository](#) möglich.
- ⇒ Für bestehende Java-Projekte ist alternativ die Integration über [Apache Maven](#) oder [Gradle](#) möglich.

Nicht (mehr) empfohlene Werkzeuge (Stand: Juli 2021): <https://swagger.io/tools/swagger-codegen/>

2.9.2 How-To: Grundlagen

Vorbedingungen:

- ⇒ Java-Runtime ist installiert und in die *PATH*-Umgebungsvariable eingebunden.
- ⇒ Das JAR-Archiv *openapi-generator-cli-*jar* wurde heruntergeladen und liegt in der entsprechenden Version lokal vor ("* = konkrete Versionsnummer; aktuell ist derzeit 5.2.0; Stand: Juli 2021)
- ⇒ Die OpenAPI-Spezifikation wurde im JSON-Format heruntergeladen (siehe oben)
→ z. B. als Datei *elster-transfer-2.1.0-rest-api-1.0-docs.json*

Komandozeilenauftruf zum Anzeigen der allgemeinen Hilfe des *OpenAPI Generator*:

⇒ `java -jar openapi-generator-cli-*jar help`

Dokumentation von *OpenAPI Generator*:

zu Optionen und Konfigurationsschaltern:

- ⇒ <https://openapi-generator.tech/docs/globals>
- ⇒ <https://openapi-generator.tech/docs/configuration>
- ⇒ zur Aufrufsyntax über die Kommandozeile: <https://openapi-generator.tech/docs/usage>
- ⇒ Übersicht über die verfügbaren Generatoren (Ausgabeformate und Programmiersprachen):
<https://openapi-generator.tech/docs/generators/README>

2.9.3 How-To: Zugriffsklassen für Client-Implementierungen generieren

Komandozeilenauftruf zum Generieren eines einfachen Clients für die Programmiersprache [Java](#):

⇒ `java -jar openapi-generator-cli-5.2.0.jar generate -i elster-transfer-2.1.0-rest-api-1.0-docs.json -g java -o myTestClient --additional-properties=library=native,dateLibrary=javas8,useJakartaEe=true,booleanGetterPrefix=is,openApiNullable=false,sourceFolder=target/generated-sources/java`

- Die Ausgabe erfolgt hier im Beispiel im Verzeichnis "myTestClient" (Option `-o`)
- Empfohlen: Angabe geeigneter [Config-Optionen](#) zur Anpassung des Generators und zur Auswahl der zu verwendenden Programmiersprachen/Frameworks. Über die

Kommandozeile werden diese Schalter über die Option "additional-properties" angegeben. Im obigen Beispiel sind das konkret:

- `library=native` zur Einstellung der verwendeten HTTP-Client- und JSON-Parser-Bibliotheken (der konkrete Wert "native" ist erst ab Java 11 möglich)
- `dateLibrary=java8` zur Steuerung, wie Zeit- und Datumsangaben dargestellt werden sollen
- `useJakartaEe=true` legt fest, dass im generierten Code die APIs u.a. für allgemeine Annotationen von [Jakarta EE](#) (Packages `jakarta.*`) statt denjenigen aus *Java EE* (Packages `javax.*`) verwendet werden sollen
(⚠ die Auswahl dieses Werts sollte zum Projektkontext passen d.h. *Jakarta EE* sollte aktiviert werden, wenn es im jeweiligen Projekt allgemein auch verwendet wird oder wenn die ältere *Java EE* zumindest nicht gleichzeitig verwendet wird)
- `booleanGetterPrefix=is` zur [Java-Bean-API](#)-konformen Benennung von generierten booleschen Getter-Methoden
- `openApiNullable=false` zur Steuerung, wie bei den generierten Methodensignaturen mit leeren (Null-)Werten umgegangen werden soll
- `sourceFolder=target/generated-sources/java` zur Erzeugung einer Apache-Maven- (bzw. Gradle)-konformen Ausgabeerzeugnis-Struktur

Die generierte Projektstruktur kann anschließend z.B. in eine Entwicklungsumgebung (Eclipse IDE, IntelliJ IDEA usw.) importiert werden. Projektdateien für *Apache Maven* und *Gradle* werden im Ausgabeordner standardmäßig ebenfalls vorgeneriert.

Die Wahl der verwendeten JSON- und HTTP-Client-Bibliotheken unter der Option "library" sollte zum jeweiligen Projektkontext passen.

- ⇒ Ab Java-Version 11 wird grundsätzlich die Verwendung des im JRE mitgelieferten HTTP-Clients per "native"-Angabe empfohlen, da dies keine zusätzliche Bibliothek für den HTTP-Client benötigt.
- ⇒ Zu beachten ist, dass für bestimmte Kombinationen der Bibliotheken Probleme zu den Code-Vorlagen des OpenAPI-Generator beschrieben sind, die zu fehlerhaft generiertem Code führen (z.B. für den Standardwert "okhttp-gson" das ⚠ [Issue 11256](#)).

Spezielle Optionen (Auswahl)

Was soll generiert werden? → kann über [globale Optionen](#) gesteuert werden:

- ⇒ `apiTests`
- ⇒ `apiDocumentation`
- ⇒ `modelTests`
- ⇒ `modelDocumentation`
- ⇒ `supportingFilesToGenerate`

Benennung der erzeugten Java-Pakete → kann über [generator-spezifische Schalter](#) gesteuert werden:

- ⇒ `apiPackage`
- ⇒ `modelPackage`
- ⇒ `invokerPackage`

Verwendung eigener Implementierung an Stelle [der vom Generator benutzten](#) Klassen:
→ kann über benutzer-definierte ["Type Mappings"](#) und ["Import Mappings"](#) gesteuert werden.

Generierung von Code für produktionsnahe Umgebungen

Je nach Konfiguration des Generators wird unterschiedlicher Code erzeugt und es können auch unterschiedliche Bibliotheken für HTTP-Clients (z.B. OkHttp, Apache Http Client, Jersey, [java.net](#)-HttpClient, REST Assured), für Darstellung von Zeit-/Datumswerten (z.B. [java.util.Date](#), Joda-Time, Java 8/JSR310), für JSON-Serialisierung (z.B. GSON, Jackson FasterXML) usw. verwendet werden. Für die Praxis sollte eine Konfiguration gewählt werden, die dem jeweiligen Einsatzzweck bzw. den gesetzten technischen Rahmenbedingungen entspricht.

- ⇒ Der generierte Code sollte in jedem Fall einem Review unterzogen werden und dabei gegen das gewünschte Einsatzszenario plausibilisiert werden. Festgestellte notwendige Ergänzungen sind zusätzlich zu programmieren.
- ⇒ Für die Übertragung größerer Datenmengen (u.a. in der Version V3 der Bereitstellung-API) sollte die Verarbeitung der jeweiligen Quelldaten nach Möglichkeit ressourceneffizient als Datenstrom (in Java z.B. über [java.util.InputStream](#)) und nicht *im Hauptspeicher* (keine Verwendung von [byte\[\]](#)-Arrays, welche die gesamten Quelldaten aufnehmen) erfolgen. Bei vielen HTTP-Clients ist dies im generierten Code leider nicht der Standard, sodass hier zusätzliche ggf. Anpassungen oder Erweiterungen erforderlich sind.

2.9.4 How-To: Mock-Server für systemübergreifende Tests generieren

Kommandozeilenauftruf zum Generieren eines Beispielprojekts für Programmiersprache/Generator [Spring](#) (unter Verwendung von [Spring Boot](#) als Framework):

- ⇒ `java -jar openapi-generator-cli-5.2.0.jar generate -i elster-transfer-2.1.0-rest-api-1.0-docs.json -g spring -o myTestServer --additional-properties=delegatePattern=true`
 - Die Ausgabe erfolgt hier im Beispiel im Verzeichnis "myTestServer" (Option `-o`)
 - Um einen Test-Server zu entwickeln, muss als Generator (Option `-g`) in jedem Fall eine [Server](#)-Implementierung ausgewählt werden, keine [Client](#)-Implementierung!
 - Empfohlen: mit Hilfe der [Config-Option](#) "delegatePattern" kann die Mock-Implementierung d.h. welche konkreten Werte im Test zurückzuliefert werden sollen, in eine separate `@Service`-Klasse verschoben werden, was die Implementierung des Test-Servers erleichtert. Über die Kommandozeile wird dieser Schalter über die Option "additional-properties" angegeben.

Die generierte Projektstruktur kann anschließend z.B. in eine Entwicklungsumgebung (Eclipse IDE, IntelliJ IDEA usw.) importiert werden. Projektdateien für [Apache Maven](#) werden im Ausgabeordner standardmäßig ebenfalls vorgeneriert.

Weitere Schritte:

- ⇒ Eine generierte Klasse namens `OpenAPI2SpringBoot` kann im gezeigten Fall verwendet werden, um den lokalen Server (als Java-Anwendung über die `main`-Methode) zu starten. Die Swagger-UI kann anschließend (analog zur "echten" ETR-Anwendung) über die Webseite <http://localhost:8081/swagger-ui.html> angerufen werden.
- ⇒ Wenn wie im o.g. Beispiel die Option "delegatePattern" verwendet wird, finden sich in der Klasse `RestApiDelegate` die vorgenerierten Beispiel-Antworten des Mock-Servers. Diese können durch die Programmierung einer eigenen, mit `@Service` annotierten Implementierung, welche die für den jeweiligen Client erforderlichen Testdaten zurückliefert, angepasst werden.

3 ELSTER-Transfer Sandbox-Variante für Entwickler

3.1 Hintergrund

Die Nutzung von ELSTER-Transfer setzt normalerweise einige Anforderungen voraus. So muss beispielsweise ein Mein ELSTER-Benutzerkonto vorhanden sein, welches für eine Organisation registriert ist. Dieses Benutzerkonto muss außerdem berechtigt sein, für ein oder mehrere Verfahren Daten auszutauschen. Den speziellen Organisationen, welche die Nutzer von ELSTER-Transfer darstellen, werden diese Berechtigungen nach einer eingehenden Prüfung gewährt.

Für Entwickler von bspw. kommunaler Software besteht keine Möglichkeit, solche Berechtigungen zu erhalten. Sie haben keine Möglichkeit, ihre Anbindung zu ELSTER-Transfer zu testen, weil sie kein Organisationszertifikat (mit einer Berechtigung) besitzen, welches Voraussetzung für die "normale" Variante von ELSTER-Transfer ist.

Dies stellte bisher kein Problem dar, da Dateien via Filesystem übergeben wurden, welche dann Mitarbeiter der speziellen Organisationen manuell oder via Batch selbst an ElsterFT/ELSTER-Transfer zum Versand übergeben haben.

Mit der Einführung der REST-API können Softwarehersteller (HKR-Hersteller, Service-Provider) erstmals die ELSTER-Transfer-Anwendung selbst steuern, ohne "Umweg" über diese Mitarbeiter.

Deshalb wird an dieser Stelle ein **separater Download für Entwickler** angeboten – eine "**Sandbox-Variante**", die der normalen Version entspricht mit folgenden wesentlichen Einschränkungen:

- ⇒ alle Verarbeitungen/Übermittlungen der Aufträge an die Systeme von ELSTER sind **fest deaktiviert**
 - ausgenommen sind lediglich Prüfung auf neue Softwareversionen und die Abfrage von allgemeinen Störungs-/Hinweismeldungen
- ⇒ beim ersten Start der Anwendung werden für alle Auftragstypen initial einige **Beispiel-Aufträge** mit verschiedenen Daten-Konstellationen (u.a. mit verschiedenen Statuswerte) in die Datenbank eingetragen. Diese können u.a. von Entwicklern verwendet werden, um z.B. Datenabholungen via REST zu testen oder die verschiedenen Bereitstellungs-Status (z. B. "Eingang bestätigt", "vom Empfänger abgeholt" ...) erproben zu können.
- ⇒ es findet keine Berechtigungsprüfung statt

3.2 Besonderheiten

Da die Sandbox-Variante primär an Entwickler gerichtet ist, unterscheidet sie sich in wesentlichen Punkten von der Variante der ELSTER-Transfer-Anwendung für die produktive Nutzung.

Wichtige Information

Die Sandbox-Variante von ELSTER-Transfer stellt lediglich eine Erleichterung für Entwickler dar, welche die Anbindung an die ELSTER-Transfer-Anwendung erproben möchten.

Sie ist nicht geeignet, um Daten mit der Steuerverwaltung auszutauschen oder Dokumente in die Postfächer von Anwendern

bereitzustellen. Jegliche Kommunikation mit den ELSTER-Servern

(ausgenommen Updateprüfung) **wurde fest deaktiviert.**

Die präsentierten Funktionalitäten (Datenaustauschverfahren mit der Finanzverwaltung; Bereitstellung von Dokumenten) der Sandbox-Variante **spiegeln nicht die tatsächlichen Berechtigungen des Nutzers wider!**

Die Datenbank der Sandbox-Variante enthält Beispielaufträge. **Es handelt sich hierbei nicht um echte Daten und auch in der Regel nicht um korrekt aufgebaute Nutzdatensätze.**

Die "normale" Variante von ELSTER-Transfer (für Nutzer) können Sie auf den Downloadseiten herunterladen: → [Lizenzvertrag und Download](#)

3.2.1 Besonderheiten bei der Installation

Unter *Linux* werden für die Sandbox-Variante zur Unterscheidung abweichende Paketbezeichner und Standard-Ordnerpfade verwendet:

	"normale" Variante für produktiven Einsatz	Sandbox-Variante
Debian-Paketname	etr	etr-sandbox
Anwendungsverzeichnis	/opt/ELSTER-Transfer/etr/	/opt/ELSTER-Transfer/etr-sandbox/
Datenverzeichnis	/home/etr/elster-transfer/	/home/etr/elster-transfer-sandbox/
Systemd-Dienstname	etr	etr-sandbox

Die während der Installation und Konfiguration durchzuführenden Schritte gelten analog, müssen aber entsprechend angepasst werden.

Unter *Docker* gelten innerhalb des Docker-Containers die regulären Ordnerpfade.

3.2.2 Besonderheiten im Verhalten der Anwendung

Aufgrund der abgeschalteten Hintergrundverarbeitung und der bereits vordefinierten Daten

- ⇒ wird ein markanter Zusatz im Titel der aufgerufenen Webseiten eingeblendet, der auf diesen Umstand hinweist
- ⇒ ist es nicht notwendig, in den "Einstellungen" der "Sandbox" ein Organisationszertifikat zu importieren
- ⇒ wird zur Nutzung nicht zwingend eine Netzwerkverbindung zu den ELSTER-Servern benötigt

3.3 Zurücksetzen einer "Sandbox"-Installation in den Initialzustand

Um z. B. zuvor erstellte Testdaten zu bereinigen ist es möglich, den Zustand einer "Sandbox"-Installation auf den Initialzustand zurückzusetzen:

- ⇒ durch Löschen des Anwendungsdatenverzeichnisses (z.B. im Rahmen einer Update-Installation)
- ETR muss beendet sein, um das Anwendungsdatenverzeichnis löschen zu können.
- ⇒ über die Funktion "Einstellungen" → "Zurücksetzen" (ab ETR-Version 24.03)

Alle bereits zuvor angelegten Aufträge und alle getätigten Änderungen gehen dabei verloren. Danach werden die bereits vordefinierten Beispiel-Aufträge (s.o.) mit dem in der Software

hinterlegten Datenstand wieder neu angelegt. Die Auftrags-IDs können dabei vom gelöschten Stand abweichen.