



Bundeszentralamt
für Steuern

Kommunikationshandbuch DIP-Standard 2.1

Standardisierte Datenübermittlung an das BZSt über die neue Massendatenschnittstelle DIP
(verfahrenübergreifende technische Beschreibung)

Dokumenten-Version: 2.1

Stand: 02/2025





Inhaltsverzeichnis

Inhaltsverzeichnis	2
Abbildungsverzeichnis	5
Tabellenverzeichnis	5
0. Informationen zum vorliegenden Dokument	6
0.1. Änderungshistorie.....	6
0.1.1. Kommunikationshandbuch (KHB).....	6
0.1.2. OpenAPI.....	7
0.1.3. XSD-Schemata.....	7
0.2. Teilnehmende Fachverfahren.....	8
0.3. Aufbau und Zweck des Dokuments.....	8
Teil 1 Allgemeine Informationen	10
1. Technische Voraussetzungen zur Nutzung der DIP-Schnittstelle	10
1.1. Systemvoraussetzungen.....	11
1.2. Bandbreite der Internetanbindung.....	11
1.3. Firewall Konfiguration.....	12
1.4. Erstellung eines Zertifikats zur Verwendung mit der DIP-Schnittstelle.....	12
1.5. Anforderungen an das Zertifikat.....	13
1.6. Schlüssel.....	13
1.7. Beispiel für die Erstellung eines privaten Schlüssels und eines Zertifikats.....	13
2. Technische Beschreibung der Kommunikation	14
2.1. Grundlagen.....	14
2.2. Authentifikation am DIP-Server.....	15
2.2.1. Anforderungstoken für die Anfrage des Access-Tokens erstellen	15
2.2.2. Access-Token anfragen.....	16
2.3. Digitale Signatur der Nutzdaten	17
2.3.1. DIP-Schnittstelle Version 1: Enveloping Signatur	18
2.3.2. DIP-Schnittstelle Version 2: Enveloped-Signatur	22
3. Freischaltung, Aktualisierung und Deaktivierung der DIP Schnittstelle	24
3.1. Freischaltung für die Massendatenschnittstelle.....	24
3.2. Aktualisierungen.....	26
3.3. Deaktivierung.....	27
4. Nutzung DIP über das Portal	27



Teil II Datenübertragung	27
5. Anforderungen an die zu übertragenden Dateien	27
5.1. Aufbau des DIP XML-Schemas Version 1	27
5.1.1. Adresse zum DIP XML-Schema.....	27
5.1.2. Aufbau des Wurzelements DIP	27
5.1.3. Aufbau des Headers	28
5.1.4. Aufbau der verfahrensspezifischen Nutzdaten.....	30
5.1.5. Beispiel	30
5.2. Aufbau des DIP XML-Schemas Version 2	31
5.2.1. Adresse zum XML-Schema	31
5.2.2. Aufbau des Wurzelements DIP	32
5.2.3. Aufbau des Headers	33
5.2.4. Aufbau der verfahrensspezifischen Nutzdaten.....	34
5.2.5. Beispiel	35
5.3. Weitere Anforderungen	36
5.3.1. Vergabe der Transferticket-ID.....	36
5.3.2. XSD-Schema Validierung.....	36
5.3.3. Encoding in UTF-8	37
5.3.4. Unzulässige Zeichen und Beschränkungen	37
5.3.5. Dateigrößenbeschränkung.....	37
6. Übertragung der Dateien	37
6.1. Aufbau der Verbindung.....	37
6.2. REST-Schnittstelle der DIP-Schnittstelle aufrufen.....	37
6.2.1. Prüfung des Access-Tokens	38
6.3. Ablauf der Datenübertragung.....	40
6.4. Abruf des Ergebnisprotokolls.....	42
6.5. REST-API der Massendatenschnittstelle	42
6.5.1. Möglicher Ablauf einer Datenlieferung im Produktivsystem.....	42
6.5.2. OpenAPI-Dokumentation.....	45
6.5.2.1. Detailbeschreibung der OpenAPI.....	46
6.5.2.2. Start einer Massendatenlieferung.....	47
6.5.2.3. Daten-XML übermitteln	48
6.5.2.4. Anhang übermitteln.....	49
6.5.2.5. Finalisierung einer Massendatenlieferung.....	50



6.5.2.6.	Abrechnen einer Massendatenlieferung	50
6.5.2.7.	Abruf des Verarbeitungsprotokolls.....	51
6.5.2.8.	Bestätigung des Abrufs des Verarbeitungsprotokolls	52
6.5.2.9.	Abruf einer Liste verfügbarer Protokolle	52
6.5.2.10.	Header	52
6.5.2.11.	Responses	53
7.	Prüfungen und Rückmeldungen der DIP-Schnittstelle	53
7.1.	Eingangsprüfungen einer DIP-Lieferung	53
7.2.	Ergebnisstatus und Fehlercodierungen	54
7.3.	Abfrage von erstellten Protokollen	57
7.4.	Aufbau des Feedback-XML-Schemas Version 1.....	58
7.4.1.	Inhalte des Elements DipResponse.....	58
7.4.2.	Beispiel	61
7.5.	Aufbau des Feedback-XML-Schemas Version 2.....	62
7.5.1.	Adresse zum XML-Schema	62
7.5.2.	Inhalte des Elements DipResponse.....	63
7.5.3.	Beispiel	65
8.	Zusätzliche Informationen	67
8.1.	Sperrungen des Zugangs zur Massendatenschnittstelle	67
8.2.	Vorgehen bei Ablauf des in der MDS hinterlegten Zertifikats	67
8.3.	Änderung des Ansprechpartners	67
9.	Abkürzungsverzeichnis/Glossar	68



Abbildungsverzeichnis

Abbildung 1 Überblick über die technischen Voraussetzungen zur Nutzung der DIP-Schnittstelle.....	10
Abbildung 2 Access-Token.....	15
Abbildung 3 Ablauf Freischaltung.....	26
Abbildung 4 Element <i>dip</i> (DipType).....	28
Abbildung 5 Element header (DipHeaderType).....	28
Abbildung 6 Element body (DipBodyType).....	30
Abbildung 7 Element <i>dip</i> (DipType).....	32
Abbildung 8 Element header (DipHeaderType).....	33
Abbildung 9 Element body (DipBodyType).....	35
Abbildung 10 REST-Zugriff mit Access-Token.....	39
Abbildung 11 Ablauf einer Datenübermittlung.....	41
Abbildung 12 Abruf von Protokollen zu MD-Lieferungen.....	58
Abbildung 13 XML-Schema <i>dipResponse</i> mit <i>consignment</i> (Teil 1).....	59
Abbildung 14 XML-Schema <i>dipResponse</i> (Teil 2).....	60
Abbildung 15 XML-Schema <i>dipResponse</i>	63
Abbildung 16 XML-Schema <i>dipResponse</i> mit <i>consignment</i> (Teil 1).....	64
Abbildung 17 XML-Schema <i>dipResponse</i> (Teil 2).....	64

Tabellenverzeichnis

Tabelle 1 Informationen zum vorliegenden Dokument.....	6
Tabelle 2 Änderungshistorie KHB.....	7
Tabelle 3 Änderungshistorie OpenAPI.....	7
Tabelle 4 Änderungshistorie XSD-Schemata.....	7
Tabelle 5 Übersicht der Umgebungen.....	12
Tabelle 6 Beschreibung der Elemente im <i>DipHeader</i>	29
Tabelle 7 Beschreibung der Elemente im <i>DipBody</i>	30
Tabelle 8 Beschreibung der Elemente im <i>DipHeader</i>	34
Tabelle 9 Beschreibung der Elemente im <i>DipBody</i>	35
Tabelle 10 URL-Präfixe in den zwei Versionen der Schnittstelle.....	38
Tabelle 11 HTTP Response Status Codes.....	40
Tabelle 12 Fehlercodes DIP.....	57
Tabelle 13 Abkürzungsverzeichnis/Glossar.....	68



0. Informationen zum vorliegenden Dokument

Dokumententitel	Kommunikationshandbuch DIP-Standard Standardisierte Datenübermittlung an das BZSt über die Massendatenschnittstelle DIP (verfahrenübergreifende technische Beschreibung)
Verantwortlicher Autor	Bundeszentralamt für Steuern
Erstellt am	28.04.2023
Zuletzt geändert am	07.02.2025

Tabelle 1 Informationen zum vorliegenden Dokument

0.1. Änderungshistorie

0.1.1. Kommunikationshandbuch (KHB)

Dokument Version	Datum	Änderung
0.1	28.04.2023	Initiale Erstellung; Bereitstellung XML-Schema
0.2	06/2023	Aktualisierung XML-Schema, Dateigrößenbeschränkung 1 Gibibyte
1.0	09/2023	Ergänzung Schnittstellendokumentation
1.1	09/2023	Änderung der URL für die Kundentestumgebung
1.2	11/2023	geänderter Ablauf zum Protokollentwurf (Abruf einer Liste mit verfügbaren Protokollen)
1.3	02/2024	Aufnahme von Erläuterungen, Überarbeitung Kapitel 2.3, Umfangreiche Ergänzung zur Open API in Kapitel 6
1.4	03/2024	Einführung einer Beschränkung zur maximalen Anzahl von Datenlieferungen innerhalb eines Zeitraums, Ergänzung der OpenAPI Spezifikation, kleiner Korrekturen
1.5	08/2024	Hinweis- und Fehlercodeergänzungen



2.0	11/2024	Xml-Schema Änderungen und damit verbundene Änderung des Signierungsverfahrens. Überarbeitung von Erläuterungen
2.1	01/2025	Darstellung der URL-Präfixe in den zwei Versionen der Schnittstelle
2.1	02/2025	Verlängerung der Umsetzungsfrist für den Wechsel auf Version 2 auf den 31.12.25

Tabelle 2 Änderungshistorie KHB

0.1.2. OpenAPI

Version	Änderung
1.0	initiale Definition der Schnittstelle
1.1	Erweiterung um Endpunkte zur Bestätigung des Protokollerhalts und zum Abruf aller Datentransfernummern mit verfügbarem Protokoll
1.2	Erweiterung der Endpunkte, welche eine Datentransfernummer erfordern, um den Returncode 424
1.3	Erweiterung des Endpunkts zum Start einer Massendatenlieferung um den Returncode 429 bei Überschreitung der Beschränkung der Anzahl erlaubter Lieferungen eines Kunden (Datenübermittlers) pro Minute
2.0	Pfad zu den Endpunkten enthält die Version (v2); Tags der Provider wurden umbenannt

Tabelle 3 Änderungshistorie OpenAPI

0.1.3. XSD-Schemata

Schema Art	Version	Änderung
DIP XSD	1.0	initiale Bereitstellung
DIP XSD	2.0	Anpassungen
DIP TYP XSD	1.0	initiale Bereitstellung
DIP TYP XSD	2.0	Anpassungen
DIP Response XSD	1.0	initiale Bereitstellung
DIP Response XSD	2.0	Anpassungen

Tabelle 4 Änderungshistorie XSD-Schemata



0.2. Teilnehmende Fachverfahren

Eine aktuelle Übersicht der teilnehmenden Verfahren finden Sie auf unserer Homepage unter

https://www.bzst.de/DE/Service/Portalinformation/Massendaten/massendaten_node.html

0.3. Aufbau und Zweck des Dokuments

Im vorliegenden Dokument wird das Verfahren zur Datenübermittlung von Massendaten über die Massendatenschnittstelle DIP an das BZSt erläutert. DIP ist die neue Massendatenschnittstelle für die Plattform OZG BZSt (PoSt). Sie ersetzt mittelfristig die bisherige Massendatenschnittstelle ELMA. Bis alle Fachverfahren die neue Schnittstelle DIP nutzen können, werden ELMA und DIP parallel betrieben.

Das Kommunikationshandbuch enthält einen standardisierten verfahrensunabhängigen Teil I, in dem die grundlegenden technischen Voraussetzungen u.a. die Freischaltung, sowie die notwendigen Softwareinstallationen, beschrieben werden. In Teil II dieses Dokumentes wird die eigentliche Datenübertragung erläutert.

Die Informationen im Teil II beziehen sich auf die Fachverfahren (vgl. Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.**), welche den in diesem Dokument beschriebenen Standardvorgaben für die DIP-Übertragung unterliegen.

Verfahrensspezifische Besonderheiten wie beispielsweise XML-Schemata sind in den jeweiligen Kommunikationshandbüchern der Fachverfahren dokumentiert.

Verfahrensspezifische Besonderheiten werden in den entsprechenden Kapiteln kenntlich gemacht.

Das DIP Kommunikationsverfahren wurde für die Übertragung von Massendaten entwickelt. Die Zielgruppe sind Großkunden, wie z. B. Rechenzentren der Kreditwirtschaft. Es werden u. a. Kenntnisse in der Datenverarbeitung und Netzwerktechnik vorausgesetzt.

Die Massendatenschnittstelle DIP stellt u.a. folgende Funktionen zur Verfügung:

- Massendatenübertragung
- Verschlüsselte Datenübertragung mit Public-Key-Authentifizierung
- Verwendung offener Standards (alle Komponenten als OpenSource verfügbar).

Zukünftig wird darüber hinaus die Möglichkeit bestehen, Massendaten, die dem DIP XML-Schema entsprechen, über eine Schnittstelle im BZSt online.Portal (Upload-Funktion) zu versenden.

Unter <https://www.bzst.de/DE/Service/Portalinformation/Massendaten/DIP/dip.html> werden folgende zusätzlichen Informationen und Hilfestellungen angeboten:



- Eine Checkliste die beim Einrichten der Schnittstelle und bei der Übermittlung von Massendaten unterstützen kann.
- Generalisierte DIP XSD Schema, DIP Type XSD und DIP Response XSD
- DIP Open API Spezifikation

Die nicht abschließende Aufzählung von Hard- und Software-Ausstattungen, Hinweise auf Seiten von Fremdanbietern und sonstige Tools liegen nicht in der Hoheit des BZSt und dienen lediglich beispielhaft der Bedienung der Massendatenschnittstelle.

Nachverfolgbarkeit von Änderungen:

Änderungen zur vorherigen Version werden wie folgt kenntlich gemacht:

- Alle geänderten Inhalte werden gelb markiert.
- Nicht mehr gültige Inhalte werden zusätzlich durchgestrichen

Hinweis: formale Änderungen werden nicht farblich markiert

Änderungen der DIP-Schnittstelle

Mit der Version 2 der DIP-Schnittstelle werden wesentliche Änderungen an der Schnittstelle vorgenommen, welche auch Auswirkungen auf bereits freigeschaltete Nutzer haben werden

1. Das zu verwendende Xml-Schema und die Rest-API wurden an 2 kleinen Stellen angepasst (Version 2).
2. Wenn Daten mit der neuen XML-Schemadefinition und der neuen API übermittelt werden (Version 2), müssen die Nutzdaten mit einer enveloped-Signatur signiert werden.

Ab dem **01.01.2025** können Massendatenlieferungen gemäß den neuen Vorgaben der xml-Schemata und der Signatur (Version 2) übermittelt werden.

Bis zum **31.12.2025** werden auch Massendatenlieferungen nach der bisherigen xml-Version und der enveloping-Signatur (Version 1) akzeptiert.

Ab dem **01.01.2026** werden nur noch Massendatenlieferungen nach den Vorgaben der Version 2 akzeptiert.

In den nachfolgenden Kapiteln werden die Änderungen jeweils mit der Überschrift „Version 1“ und „Version 2“ kenntlich gemacht. Ein Umstieg im ersten Halbjahr 2025 auf die Version 2 wird empfohlen.



Teil 1 Allgemeine Informationen

Die nachfolgenden Abschnitte des Teils I - Allgemeine Informationen - gelten für alle an DIP angebotenen Fachverfahren und bieten einen Überblick über die technischen Voraussetzungen zur Nutzung der Massendatenschnittstelle DIP (DIP-SST).

1. Technische Voraussetzungen zur Nutzung der DIP-Schnittstelle

Über die Schnittstelle des Digitalen Posteingangs (DIP) können Kunden Massendaten an Fachverfahren des BZSt weiterleiten. Die technischen Voraussetzungen werden in diesem Abschnitt beschrieben. **Der allgemeine Ablauf wird in folgender Grafik dargestellt. Die Fehlerbehandlung wurde zur Vereinfachung weggelassen. Die Grafik dient als Überblick. Die einzelnen Schritte werden in den nachfolgenden Kapiteln beschrieben.**

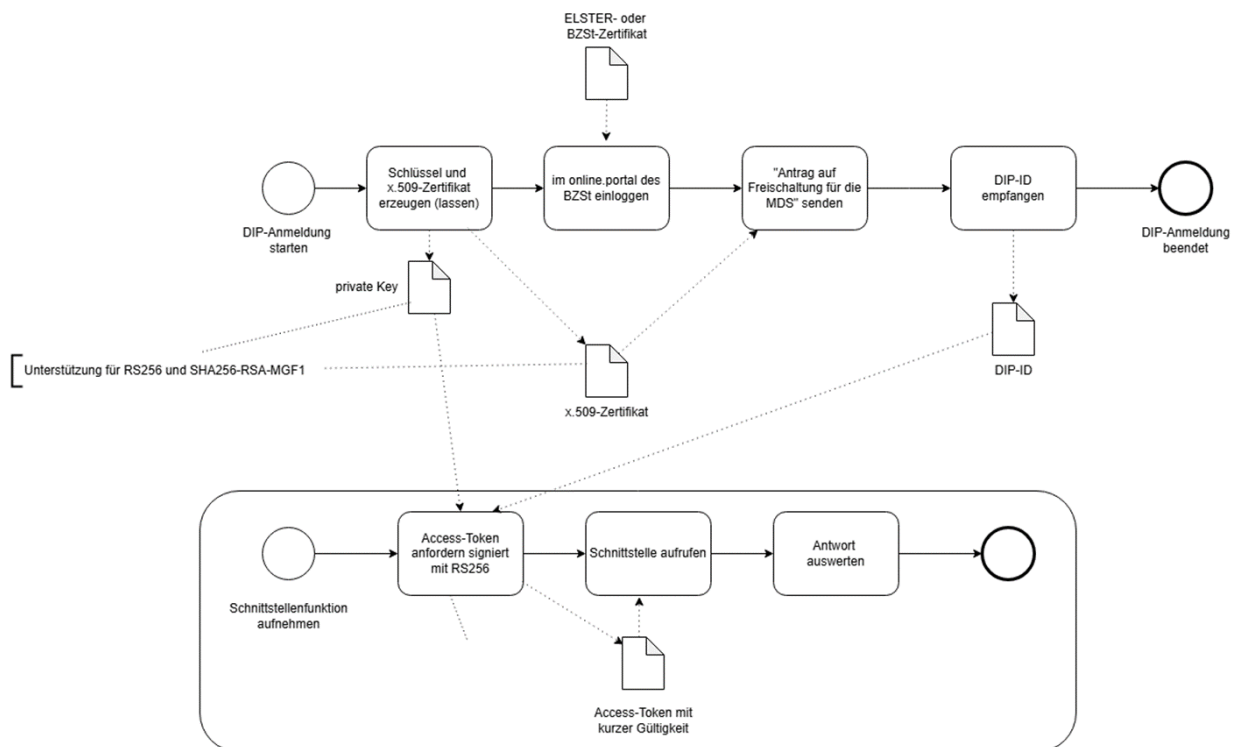


Abbildung 1 Überblick über die technischen Voraussetzungen zur Nutzung der DIP-Schnittstelle

Der obere Prozess zeigt den Ablauf zur Freischaltung der DIP-Schnittstelle. Um die Schnittstelle nutzen zu können, muss ein Zertifikat (und zudem ein privater Schlüssel) erstellt werden, welche die geforderten Algorithmen unterstützen. Für die eigentliche Freischaltung der DIP-Schnittstelle ist ein Antrag über <https://online.portal.bzst.de/> zu



stellen (Formular: "Antrag zur Freischaltung für die Massendatenschnittstelle"). In diesem Formular wird das zuvor erzeugte Zertifikat an die DIP-Schnittstelle übergeben. Zur Nutzung des BZSt online Portals ist eine Anmeldung an dem Portal mit dem entsprechenden Zugangszertifikat des Portals (z.B. Elster oder BZSt-Zertifikat) notwendig. Hierbei handelt es sich nicht um das zuvor erstellte und für die Freischaltung der Massendatenschnittstelle benötigte Zertifikat. Ein ELSTER oder BZSt-Zertifikat zur Anmeldung am BZSt online Portal sollte bereits vorliegen. Als Antwort auf den Freischaltungsantrag wird eine DIP-ID für die DIP-Schnittstelle erzeugt und an den Antragsteller per E-Mail übermittelt.

Für den Aufruf der einzelnen Endpunkte der DIP-Schnittstelle muss ein Access-Token beantragt werden. Dies erfolgt über ein mit dem privaten Schlüssel signiertes Requesttoken in dem auch die DIP-ID steht. Für die Signierung wird der für den aktuellen Stand der Anmeldung / Aktualisierung relevante Algorithmus verwendet. Mit dem nur kurz gültigen Requesttoken kann dann der Endpunkt aufgerufen werden.

1.1. Systemvoraussetzungen

Um die Massendatenschnittstelle von DIP nutzen zu können, wird die folgende Software benötigt:

- Ein Programm zur Dateiübertragung via HTTPS
 - Notwendig für die Interaktion mit der bereitgestellten Schnittstelle und Übertragung einer Massendaten-Lieferung.
 - Dies ist im Allgemeinen das Backend-System des Kunden, das die Kommunikation regelmäßig startet. Tools wie beispielsweise cURL können ebenfalls für eine (Batch)-Verarbeitung eingesetzt werden, um die im Kapitel [6 Übertragung der Dateien](#) beschriebenen Requests abzusetzen.
- X.509 Public-Key-Zertifikat; die Anforderungen werden in dem Kapitel [1.5 Anforderungen an das Zertifikat](#) beschrieben
- OpenSSL
 - zur Erstellung eines Schlüsselpaares und eines Nutzerzertifikats, wenn der Kunde diese selbst erstellen möchte.

1.2. Bandbreite der Internetanbindung

Die DIP-Schnittstelle kann Datenlieferungen bis zu einem Gigabyte entgegennehmen. Die Dauer eines Requests und damit auch die Zeit für einen Upload über die Schnittstelle ist auf 15 Minuten begrenzt. Es ist sicherzustellen, dass die liefernden Systeme mit einer ausreichenden Bandbreite angebunden sind. Andernfalls müssen die Lieferungen in kleinere Teillieferungen aufgeteilt werden.



1.3. Firewall Konfiguration

Bei der Verwendung einer Firewall im Kundensystem ist beim Sender die Freischaltung der IP-Adresse des DIP-Servers mit dem Port 443 zu konfigurieren, um die Nutzung von HTTPS zu ermöglichen.

Die Massendatenschnittstelle DIP wird in den folgenden Umgebungen bereitgestellt:

Name der Umgebung	Beschreibung / Zweck	URL
MDS-PROD	Bezeichnung der Umgebung, auf welcher die DIP-Schnittstelle zur produktiven Verwendung durch die Nutzenden bereitgestellt wird	https://mds.bzst.bund.de
MDS-KTST	Bezeichnung der Umgebung, welche den Nutzenden die Möglichkeit des Versands von Testdaten bietet, um die Interaktion mit der DIP-Schnittstelle zu testen	https://mds-ktst.bzst.bund.de

Tabelle 5 Übersicht der Umgebungen

1.4. Erstellung eines Zertifikats zur Verwendung mit der DIP-Schnittstelle

Für die Freischaltung und die Datenlieferung an die DIP-Schnittstelle wird je Kunde ein Zertifikat benötigt. Es können alle X.509 Public-Key-Zertifikate verwendet werden, die automatisch geprüft werden können. Aussteller können z.B. sein

- Zertifikate von einer offiziellen Zertifizierungsstelle
- Self-Signed-Zertifikate: Zertifikate, die der Kunde selbst erstellt.

Hinweise:

Die Zertifikate zur Anmeldung am BZSt online.portal sind hier **nicht** zu verwenden!

■ Algorithmen für die Signierung

Mit dem Schlüssel muss sowohl mit SHA256-RSA-MGF1 (bei der Datenlieferung) als auch mit RS256 (bei der Anfrage des Access-Tokens) signiert werden können. Es muss daher sichergestellt werden, dass der erzeugte Schlüssel nicht auf einen der beiden Algorithmen beschränkt ist. Bei der Verwendung von OpenSSL bedeutet dies, dass einige der *-pkeyopt* Optionen nicht verwendet werden sollten, da diese die Nutzung des Schlüssels beschränken.



1.5. Anforderungen an das Zertifikat

Das Zertifikat muss im „pem-Format“ abgelegt werden. Folgende Anforderungen werden an die Inhalte des Zertifikats gestellt:

- CN (Common Name) - Domäne des aufrufenden Kundensystems (des Datenübermittlers)
- EMAILADDRESS (Emailadresse)
- O (Organisation)
- L (Location, Stadt)
- ST (State, Bundesland)
- C (Country, Land)

Als Signaturalgorithmus wird RSASSA-PSS1 mit folgenden Parametern verwendet:

- Hashverfahren: SHA-256
- Mask Generation Function: MGF1 mit SHA-256
- Länge des Salts: 32 Byte
- Trailer Field: 0xBC

Die vorgegebenen Parameter entsprechen den Vorgaben des Signatureschemas RSASSA-PSS, wie es in RFC 8017 beschrieben wird. Die Definition der Standard-Parameter nach RFC 4055, bei welcher das Trailer Field auf den Wert 0x01 festgelegt wurde, wird ebenfalls unterstützt.

1.6. Schlüssel

Es wird eine Schlüssellänge von mindestens 4096 Bit erwartet.

1.7. Beispiel für die Erstellung eines privaten Schlüssels und eines Zertifikats

Ein Beispiel für die Erzeugung – dies ist keine Vorgabe oder Empfehlung

Um einen privaten Schlüssel und ein Zertifikat zu erzeugen, kann wie im Folgenden vorgegangen werden. Dies ist **keine** Empfehlung, sondern soll nur das Verfahren allgemein illustrieren. Insbesondere müssen Schlüssellängen selbst definiert werden, diese dürfen aber 4096 Bit nicht unterschreiten. Im Allgemeinen würde auch eine CA zum Einsatz kommen und ein Passwort für den privaten Schlüssel vergeben werden. Ebenso muss eigenverantwortlich die Gültigkeitsdauer des Zertifikats festgelegt werden, sodass Datenschutzgesichtspunkte beachtet werden.

```
openssl req -newkey rsa-pss -new -nodes -x509 -days 3650 -pkeyopt rsa_keygen_bits:4096 -sigopt rsa_pss_saltlen:32 -keyout key.pem -out cert.pem
```



Wenn der öffentliche Schlüssel benötigt wird, kann er extrahiert werden

```
openssl x509 -pubkey -noout -in cert.pem > pubkey.pem
```

Das erzeugte Zertifikat hat folgende Werte, die die Anforderungen von oben widerspiegeln (z.B. ein salt von hex 0x20).

Werte des Zertifikats

```
$ openssl x509 -in cert.pem -text
```

```
Certificate:
```

```
    Data:
```

```
        Version: 3 (0x2)
```

```
        Serial Number:
```

```
            3e:20:c3:42:1e:a4:89:0e:f2:7f:1b:e1:5f:b2:26:15:13:46:45:9f
```

```
        Signature Algorithm: rsassaPss
```

```
            Hash Algorithm: sha256
```

```
            Mask Algorithm: mgf1 with sha256
```

```
                Salt Length: 0x20
```

```
            Trailer Field: 0xBC (default)
```

```
        Issuer: ...
```

```
...
```

■ Hinweis zur Zertifikatserstellung

Bei der Erstellung eines Zertifikats unter Nutzung des Tools OpenSSL ist darauf zu achten, dass in manchen Versionen (beispielsweise 1.0.x und 3.1.x) für das Signaturschema RSASSA-PSS die Standard-Parameter nach RFC 4055 verwendet werden. Hierbei weicht der Wert des Trailer Field von dem geforderten Wert ab, wird dennoch bei der Validierung des Zertifikats unterstützt.

2. Technische Beschreibung der Kommunikation

2.1. Grundlagen

Die Übertragung der Massendaten erfolgt über Aufrufe von REST-Endpunkten. Die Verbindung muss auf Basis von **HTTPS** aufgebaut werden. Des Weiteren muss für den Verbindungsaufbau ein Access-Token bei der DIP-Schnittstelle erfragt werden, welches über ein Anforderungstoken beantragt wird. Die Signierung muss auf Seite des Kunden mit dem privaten Schlüssel erfolgen, der auch für die Erzeugung des Zertifikats bei der Freischaltung genutzt wurde. Die Prüfung des Tokens erfolgt in der DIP-Schnittstelle über eben dieses Zertifikat. Die gesendeten Massendaten müssen mit dem gleichen Schlüssel signiert werden, der für die Beantragung des Access-Tokens verwendet wird.



2.2. Authentifikation am DIP-Server

Bei der Kommunikation wird im ersten Schritt ein Access-Token beantragt. Hierfür sendet das Kundensystem ein Anforderungstoken an die DIP-Schnittstelle.

Access-Token für den Aufruf der REST-Schnittstelle anfordern

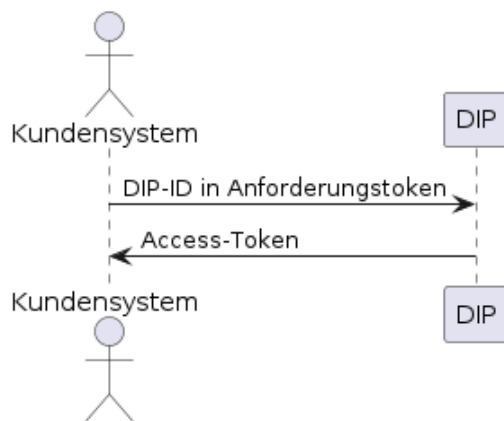


Abbildung 2 Access-Token

Nach Erhalt des Access-Tokens kann ein REST-Service von DIP aufgerufen werden. Der Ablauf der einzelnen Aufrufe für eine Datenlieferung an die DIP-Schnittstelle ist in Kapitel 6 [Übertragung der Dateien](#) beschrieben.

2.2.1. Anforderungstoken für die Anfrage des Access-Tokens erstellen

Bei der Massendatenschnittstelle werden Anforderungen für Accesstoken mit dem Algorithmus RSA256 signiert.

Ein Anforderungstoken ist ein signiertes *JSON Web Token* (JWT). Folgende Informationen müssen im JWT für die Anfrage hinterlegt sein:

- Issuer: DIP-ID des Kunden, welche bei der Freischaltung vergeben wurde
- Audience:
 - Es wird die URL der jeweiligen Umgebung verwendet (siehe Kapitel 1 [Technische Voraussetzungen zur Nutzung der DIP-Schnittstelle](#)) erweitert um den Pfad **/auth/realm/mds**
- Subject hat den gleichen Wert wie Issuer - also ebenfalls die DIP-ID
- Issued at: aktueller Zeitpunkt
- Expires At: Ablaufzeitpunkt des Anforderungstokens
- JWT-ID: eindeutige ID des JWT - kann durch den Kunden frei vergeben werden
- Not Before: ab diesen Zeitpunkt ist das Token frühestens gültig (es kann ggf. Zeitdifferenzen zwischen den Systemen geben)



Nach Erstellen des Anforderungstokens muss dieses signiert werden (im Beispiel wird dies mit Aufruf von *sign* realisiert). Die Anforderungen an die Signatur sind in Kapitel 1 [Technische Voraussetzungen zur Nutzung der DIP-Schnittstelle](#) dokumentiert.

Ein Beispiel für die Erstellung eines Anforderungstokens für die Produktionsumgebung in an Java angelehnten Pseudocode könnte wie folgt aussehen:

```
JWT.create()  
    .withIssuer("936DA01F-9ABD-4D9D-80C7-02AF85C822A8") // Client ID  
    .withSubject("936DA01F-9ABD-4D9D-80C7-02AF85C822A8") // Client ID  
    .withAudience("https://mds.bzst.bund.de/auth/realms/mds") // der Realm  
    .withIssuedAt(new Date()) // Anforderungszeitpunkt  
    .withExpiresAt(new Date(System.currentTimeMillis() + (5 * 60 * 1000L))) // 5 min  
    .withJWTid(UUID.randomUUID().toString()) // eindeutige ID  
    .withNotBefore(new Date(System.currentTimeMillis() - 60 * 1000L)) // z.B. 1 Minute  
    vor dem aktuellen Zeitstempel  
    .sign(Algorithm.RSA256((RSAPublicKey) getPublicKey(), (RSAPrivateKey)  
getPrivateKey())); // Signieren mit dem privaten Schlüssel
```

Für die Kundentestumgebung muss die URLA in dem Feld „withAudience“ auf die Kundentestumgebung gesetzt werden (<https://mds-ktst.bzst.bund.de/auth/realms/mds>).

■ Anmerkungen zu den Algorithmen

⚠ Für die Erstellung des Zertifikats für die Freischaltung und bei der Signierung des JWT werden **unterschiedliche** Algorithmen verwendet.

Das Zertifikat für die Registrierung bei DIP und implizit auch die Public/Private Keys müssen mit dem **RSA-PSS** Algorithmus erzeugt werden.

Im Algorithmus abweichend wird das JWT Token mit dem erzeugten Private Key (basierend auf RSA-PSS) signiert, allerdings muss hier der Algorithmus **RS256** verwendet werden (dieser wird in einigen Java Bibliotheken als RSA256 bezeichnet). **Bei der Signierung der Massendatennachrichten wird dann SHA256-RSA-MGF1 als Variante von PSS verwendet.**

2.2.2. Access-Token anfragen

Um ein Access-Token zu erhalten, muss das erzeugte Anforderungstoken an die Schnittstelle gesendet werden. Hierfür wird ein **POST-Request** an die URL der DIP-Schnittstelle gesendet. Der Content-Type ist application/x-www-form-urlencoded und es müssen folgende Felder vorhanden sein:

- grant_type: fester Wert "client_credentials"
- client_assertion_type: fester Wert "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
- client_assertion: das signierte JWT als Base64
 - ⚠ Einige Bibliotheken erzeugen automatisch eine Base64 Kodierung. Achten Sie darauf, dass nicht eine doppelte Base64 Kodierung erfolgt.



Für den Zugriff in den verschiedenen Umgebungen wird die Umgebungs-URL (siehe Kapitel 1 [Technische Voraussetzungen zur Nutzung der DIP-Schnittstelle](#)) mit dem Endpunkt `"/auth/realms/mds/protocol/openid-connect/token"` verwendet. Der "Assertion" signierte JWT kann nur einmal benutzt werden. Wenn eine zweite Anfrage notwendig ist, muss es eine neue JWT generiert werden (Schritt *"Anforderungstoken für die Anfrage des Access-Tokens erstellen"* neu durchführen).

⚠ Hinweise zur JWT Nutzung:

- Das signierte JWT in dem Feld "client_assertion" kann nur für eine Anfrage benutzt werden.
- Für weitere Anfragen muss jeweils ein neues JWT generiert werden (Schritt *"Anforderungstoken für die Anfrage des Access-Tokens erstellen"* neu durchführen).
- Das erzeugte JWT kann durch Online oder Offline-Tools geprüft werden. Bei Problemen mit dem JWT, sollte der Inhalt des erzeugten JWT als erstes geprüft werden.

Beispiel einer Anfrage für ein Access-Token in der Produktionsumgebung

```
POST https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=openid&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer&client_assertion=<hier das signierte JWT>
```

Beispiel einer Anfrage für ein Access-Token in der Kundentestumgebung

```
POST https://mds-ktst.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=openid&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer&client_assertion=<hier das signierte JWT>
```

2.3. Digitale Signatur der Nutzdaten

Jede Massendaten-Lieferung, die über die DIP-Schnittstelle übertragen werden soll, muss durch den Kunden mit SHA256-RSA-MGF1 (analog zu PS256) signiert werden. Hierbei unterscheiden sich die Versionen 1 und 2 der DIP-Schnittstelle. In Version 1 wird eine Enveloping-Signatur verwendet. In Version 2 eine Enveloped-Signatur.

■ Datensignierung Version 1

Enveloping-Signatur



■ Datensignierung Version 2

Enveloped-Signatur

■ Signierte Daten dürfen nicht verändert werden

Nach der Signierung muss der String unverändert bleiben, sonst wird die Signatur ungültig. Ein Kunde hatte das Problem, dass bei der Übertragung des Strings Veränderungen erfolgten (vermutlich Formatierungen). Eine Übertragung als Bytestream zeigte dieses Verhalten nicht.

Die digitale Signatur wird durch den Benutzer mit seinem privaten Schlüssel, der auch für die Erzeugung des Zertifikats für den Freischaltungsantrag diente, erstellt. Sie beweist die Urheberschaft des Kundensystems und stellt die Integrität der Massendaten-Lieferung sicher.

CanonicalizationMethod

Da es sich um XML handelt, wird eine Kanonisierung (XML-Normalisierung) oder kurz C14 auf die Daten vor der Signierung angewendet. Hierdurch wird eine standardisierte Darstellung des XMLs erreicht (z.B. bezüglich Leerzeichen, Zeilenenden, ...). In dem Feld wird die Methode, welche zum Konvertieren des Objekts in ein XML-Element in kanonischer Form verwendet wird, angegeben.

⚠ Es werden **ausschließlich** die folgenden Varianten der c14n Kanonisierung bei der Validierung der Signatur durch die DIP-Schnittstelle unterstützt:

- <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- <http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments>
- <http://www.w3.org/2001/10/xml-exc-c14n#>
- <http://www.w3.org/2001/10/xml-exc-c14n#WithComments>
- <http://www.w3.org/2006/12/xml-c14n11>
- <http://www.w3.org/2006/12/xml-c14n11#WithComments>

2.3.1. DIP-Schnittstelle Version 1: Enveloping Signatur

Zur Signierung der zu sendenden Daten wird das gesamte dip-Element mit einer **Enveloping-Signatur** signiert. Das bedeutet, dass sich die Signatur und das dip-Element in der gleichen XML-Datei befinden und die Signatur das dip-Element umschließt. Die Signatur ist also in der zu übertragenden Datei integriert, indem die Signatur als Wrapper um das Daten-XML herumgelegt wird. Im Folgenden betrachten wir die hier dargestellte Massendatenlieferung:

Beispiel-XML: Unsigniertes dip-Element Version 1

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<dip xmlns="http://itzbund.de/ozg/bzst/post/dip/v1/" version="1.0">
  <header environment="TEST">
    ...
  </header>
  <body>
    ...
  </body>
</dip>
```

Bei der Signierung mit der Enveloping-Signatur wird das dip-Element als Kindknoten des ds:Objects eingehangen und mit diesem gemeinsam signiert.

Das Signature-Element enthält hierbei neben dem dip-Element folgende Informationen, mit denen die Integrität des dip-Elements überprüft wird:

- Reference\URI: Verweis auf die ID des signierten Elements
 - bezieht sich im Kontext der Massendatenschnittstelle auf das ds:Object-Element, welches das dip-Element beinhaltet.
 - Bei der Angabe der ID ist eine Raute voranzustellen:
 - <ds:Reference URI="#object">
- CanonicalizationMethode
- DigestMethod: Bezeichner der Methode, welche zum Berechnen des DigestValues aus der kanonischen Form des signierten Elements verwendet wird
- DigestValue: Base64-Hashwert des signierten Elements gemäß der verwendeten DigestMethod
- SignatureMethod: Kennzeichner des Algorithmus, der zur Berechnung des SignatureValues aus der kanonischen Form des SignedInfo-Elements (beinhaltet den DigestValue) verwendet wird
- SignatureValue: Wert der Signatur, der die SignedInfo (beinhaltet den DigestValue) signiert
 - erfordert bei der Erstellung den privaten Schlüssel des Kunden, der auch für die Erzeugung des Zertifikats bei der Freischaltung genutzt wurde.
- X509Certificate: beinhaltet das Zertifikat mit dem öffentlichen Schlüssel des Kunden
 - enthält dasselbe Zertifikat, welches bei der Freischaltung genutzt wurde.
 - Der angegebene öffentliche Schlüssel des Kunden wird zur Verifizierung der Gültigkeit des SignatureValues durch die DIP-Schnittstelle verwendet.

Auf diese Weise entsteht aus der oben stehenden beispielhaften XML-Datei mit dem unsignierten dip-Element durch das Hinzufügen der Enveloping-Signatur die folgende XML-Datei:

Beispiel-XML: Signierte Datei mit Enveloping-Signatur (DSig), die das dip-Element enthält

```
<?xml version="1.0" encoding="UTF-8"?>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```



```
<ds:SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-  
MGF1"/>  
<ds:Reference URI="#object">  
  <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>  
  <ds:DigestValue><!-- base64-codierter Digest-value einfügen --></ds:DigestValue>  
</ds:Reference>  
</ds:SignedInfo>  
<ds:SignatureValue><!-- base64-codierte Signatur einfügen --></ds:SignatureValue>  
<ds:KeyInfo>  
  <ds:X509Data>  
    <ds:X509SubjectName><!-- Subject Name einfügen --></ds:X509SubjectName>  
    <ds:X509Certificate><!-- base64-codiertes Zertifikat einfügen --></ds:X509Certificate>  
  </ds:X509Data>  
</ds:KeyInfo>  
<ds:Object Id="object">  
  <dip xmlns="http://itzbund.de/ozg/bzst/post/dip/v1/" version="1.0">  
    <header environment="TEST">  
      ...  
    </header>  
    <body>  
      ...  
    </body>  
  </dip>  
</ds:Object>  
</ds:Signature>
```

Durch das Prüfen der Stimmigkeit dieser Werte und dem Vergleich des in der Signatur angegebenen Zertifikats mit dem in der Benutzerverwaltung hinterlegten Zertifikat kann verifiziert werden, dass das dip-Element vom Inhaber des Private Keys des zugehörigen bei der Anmeldung zur Massendatenschnittstelle angegebenen Zertifikats erstellt und nicht im Nachhinein manipuliert wurde.

■ Eine Teilimplementierung in JAVA

Eine Implementierung der Signierung in JAVA unter Verwendung von Klassen aus dem Bereich javax.xml.crypto, javax.xml.parsers und javax.xml.transform könnte wie folgt aussehen. Dieser Code soll nur das Vorgehen beschreiben und ist keine Implementierungsvorgabe oder -empfehlung.

Signieren des XML – Vorgehensbeispiel in JAVA

```
...  
InputStream unsignedDipStream  
= new ByteArrayInputStream(dipXml.getBytes(StandardCharsets.UTF_8));  
  
String digestMethod = DigestMethod.SHA256;  
  
String signatureMethod = SignatureMethod.SHA256_RSA_MGF1;
```



```
var documentBuilderFactory = DocumentBuilderFactory.newInstance();

documentBuilderFactory.setNamespaceAware(true);

Document dipXmlDocument =
documentBuilderFactory.newDocumentBuilder().parse(unsignedDipStream);

Document newDocument = documentBuilderFactory.newDocumentBuilder().newDocument();

XMLSignatureFactory xmlSignatureFactory = XMLSignatureFactory.getInstance("DOM");

var domSignContext = new DOMSignContext(getPrivateKey(), newDocument);

domSignContext.setDefaultNamespacePrefix("ds");

var reference = xmlSignatureFactory.newReference("#object",
xmlSignatureFactory.newDigestMethod(digestMethod, null), List.of(), null, null);

var content = new DOMStructure(dipXmlDocument.getDocumentElement());

var signedObject =
xmlSignatureFactory.newXMLObject(Collections.singletonList(content), "object", null, null); // Start
der Signierung bei object

var signedInfo = xmlSignatureFactory.newSignedInfo(

    xmlSignatureFactory.newCanonicalizationMethod(CanonicalizationMethod.INCLUSIVE,
(C14NMethodParameterSpec) null),
    xmlSignatureFactory.newSignatureMethod(signatureMethod, null),
Collections.singletonList(reference));

var keyInfo = keyInfoAusZertifikatErzeugen(xmlSignatureFactory, testzertifikatErmitteln());

var xmlSignature = xmlSignatureFactory.newXMLSignature(signedInfo, keyInfo,
Collections.singletonList(signedObject), null, null);

xmlSignature.sign(domSignContext);

Source xmlSource = new DOMSource(newDocument);

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

Result outputTarget = new StreamResult(outputStream);

TransformerFactory.newInstance().newTransformer().transform(xmlSource, outputTarget);
```



```
return new ByteArrayInputStream(outputStream.toByteArray()); // Rückgabe eines InputStream mit dem signierten XML
```

2.3.2. DIP-Schnittstelle Version 2: Enveloped-Signatur

Bei der Signierung mit der Enveloped-Signatur wird die gesamte Nachricht signiert. Das Signature-Element enthält hierbei folgende Informationen, mit denen die Integrität des dip-Elements überprüft wird:

- Reference\URI: Leerstring
- CanonicalizationMethode
- DigestMethod: Bezeichner der Methode, welche zum Berechnen des DigestValues aus der kanonischen Form des signierten Elements verwendet wird
- DigestValue: Base64-Hashwert des signierten Elements gemäß der verwendeten DigestMethod
- SignatureMethod: Kennzeichner des Algorithmus, der zur Berechnung des SignatureValues aus der kanonischen Form des SignedInfo-Elements (beinhaltet den DigestValue) verwendet wird
- SignatureValue: Wert der Signatur, der die SignedInfo (beinhaltet den DigestValue) signiert
 - erfordert bei der Erstellung den privaten Schlüssel des Kunden, der auch für die Erzeugung des Zertifikats bei der Freischaltung genutzt wurde.
- X509Certificate: beinhaltet das Zertifikat mit dem öffentlichen Schlüssel des Kunden
 - enthält dasselbe Zertifikat, welches bei der Freischaltung genutzt wurde.
 - Der angegebene öffentliche Schlüssel des Kunden wird zur Verifizierung der Gültigkeit des SignatureValues durch die DIP-Schnittstelle verwendet.

Eine Teilimplementierung in Java

Eine Implementierung der Signierung in JAVA unter Verwendung von Klassen aus dem Bereich *javax.xml.crypto*, *javax.xml.parsers* und *javax.xml.transform* könnte wie folgt aussehen. Dieser Code soll nur das Vorgehen beschreiben und ist **keine** Implementierungsvorgabe oder -empfehlung.

Signieren des XML mit Enveloping-Signatur- Vorgehensbeispiel in JAVA

```
...  
InputStream unsignedDipStream  
  
String digestMethod  
  
String signatureMethod
```



```
Document dipXmlDocument =
getDipXmlDocumentAusStream(DocumentBuilderFactory.newInstance(), unsignedDipStream);

XMLSignatureFactory xmlSignatureFactory = XMLSignatureFactory.getInstance("DOM");

DOMSignContext domSignContext = new DOMSignContext(getPrivateKey(),
dipXmlDocument.getDocumentElement());

List<Transform> transformList =
Collections.singletonList(xmlSignatureFactory.newTransform(Transform.ENVELOPED,
(TransformParameterSpec) null));

SignedInfo signedInfo = getSignedInfo(xmlSignatureFactory, "", digestMethod, transformList,
signatureMethod);

KeyInfo keyInfo = keyInfoAusZertifikatErzeugen(xmlSignatureFactory, testzertifikatErmitteln());

XMLSignature xmlSignature = xmlSignatureFactory.newXMLSignature(signedInfo, keyInfo);

xmlSignature.sign(domSignContext);

return getOutputDocumentAsStream(dipXmlDocument);
```

Eine signierte Datei - hier für die Lesbarkeit als "Pretty Print" - könnte wie folgt aussehen:

Signierte Daten der Version 2 mit Enveloped-Signatur

```
<?xml version="1.0" encoding="UTF-8"?>
<dip xmlns="http://itzbund.de/ozg/bzst/post/dip/v2/" version="2.0">
  <header environment="PROD">
    <consignment>
      <customerIdentifier>
        <identityProvider>...</identityProvider>
        <identifier>...</identifier>
      </customerIdentifier>
      <creationTime>2023-04-01T10:11:22</creationTime>
      <transferticketId>5555-5556665-6464765764_56ZGHUVMN</transferticketId>
    </consignment>
    <application code="CESOP"/>
  </header>
  <body>
    <consignmentItem consignmentItemPosition="42">
      <data>
        <cesop:CESOP xmlns:cesop="urn:ec.europa.eu:taxud:fiscalis:cesop:v1" version="4.0">
          ...
        </cesop:CESOP>
      </data>
    </consignmentItem>
```



```
</body>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-
MGF1"/>
    <Reference URI="">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <DigestValue>w0n9WkbDlxXxP2x+mzfZ+y5Bsl6A5gacdOfCZXTZ1/c=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue><!-- base64-codierte Signatur einfügen --></SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509SubjectName>L=Muelheim-Kaerlich,ST=Rheinland-Pfalz,C=DE</X509SubjectName>
      <X509Certificate><!-- base64-codiertes Zertifikat einfügen --></X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>
</dip>
```

3. Freischaltung, Aktualisierung und Deaktivierung der DIP Schnittstelle

Über die DIP-Schnittstelle ist es möglich, Daten zwischen beliebigen Endstellen und dem ITZBund als Dienstleister für das BZSt zu übertragen. Für diese Kommunikation werden eine erfolgreiche Registrierung bei dem BZSt [online.portal](http://www.online.portal.bzst.de) und die Freischaltung für das DIP-Verfahren vorausgesetzt.

Sowohl für das Produktivsystem, als auch für die Kundentestumgebung ist eine Freischaltung erforderlich.

3.1. Freischaltung für die Massendatenschnittstelle

Zur Kommunikation mit der DIP-Schnittstelle wird eine explizite Freischaltung benötigt. Die Beantragung der Freischaltung erfolgt über das [BZSt online.portal](http://www.online.portal.bzst.de) (www.online.portal.bzst.de) mit dem Formular "**Antrag auf Freischaltung für die Massendatenschnittstelle BZSt online.portal (DIP)**".



Für die Anmeldung am BZSt online.portal ist ein ELSTER- oder eine BZSt-Zertifikat Ihrer Organisation erforderlich.

Durch die Anmeldung am Portal ist eine eindeutige Zuordnung zu einem Ordnungsmerkmal über das bei der Anmeldung verwendete Zertifikat gegeben. Als Ordnungsmerkmal werden

- die Steuernummer (bei der Identifizierung im Portal über Elster) oder
- die BZST-Nummer (bei der Identifizierung im Portal über „BZSt“

verwendet. Diese Daten werden automatisch aus der Anmeldung am Portal in dem Formular gesetzt und müssen nicht manuell befüllt werden. Zu einem Ordnungsmerkmal kann es nur einen Zugang zur DIP-Schnittstelle geben. Findet eine Anmeldung mit einem Elster-Zertifikat statt, ist das Ordnungsmerkmal die Steuernummer. Erfolgte die Anmeldung mit einem BZSt-Zertifikat, ist das Ordnungsmerkmal die BZST-Nummer.

■ Hinweis Ordnungsmerkmal

Da der Zugang an das Ordnungsmerkmal geknüpft ist, muss der Kunde sicherstellen, dass bei Änderung oder Deaktivierung des Erstantrags bzgl. der Freischaltung der DIP-Schnittstelle, immer die gleiche Form der Anmeldung des durchführenden Portalnutzers erfolgt.

In dem Formular für die Anmeldung werden alle notwendigen Informationen erfasst, die für die Nutzung der MDS notwendig sind:

- Ordnungsmerkmal - vorausgefüllt und nicht veränderbar
 - Firmendaten - vorausgefüllt und nicht veränderbar
 - technischer Ansprechpartner (an diesen wird, zusätzlich zum Portalnutzer, eine Information zur Freischaltung bzw. Aktualisierung gesendet)
 - das für die Authentifizierung der Datenübertragung und die Signaturprüfung der Massendaten zu verwendende Zertifikat im **pem**-Format, welches die in Kapitel „1 Technische Voraussetzungen zur Nutzung der DIP-Schnittstelle“ beschriebenen Anforderungen erfüllt.
- ⚠ Es muss hier ein Zertifikat und **nicht** der öffentliche Schlüssel hochgeladen werden.

Im folgenden Diagramm ist der Ablauf einer erfolgreichen Freischaltung für die Massendatenschnittstelle dargestellt:

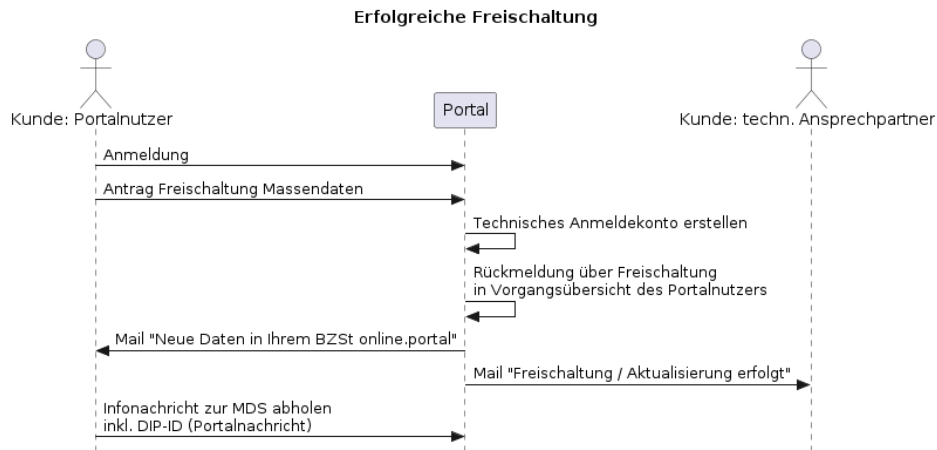


Abbildung 3 Ablauf Freischaltung

Schritt 1: Anmeldung am BZSt online.portal

Schritt 2: Beantragung der Freischaltung zur Nutzung der MDS

Schritt 3: Anlegung eines technischen Anmeldekontos und Speicherung der erhaltenen Informationen (inklusive des Zertifikats) durch das Portal

Schritt 4: Rückmeldung über erfolgreiche Freischaltung und nötige Zugangsdaten werden in Form einer Infonachricht in der Vorgangsübersicht des Antragstellers (Portalnutzer) durch das Portal eingestellt

Schritt 5: Versand der Benachrichtigung zum Erhalt neuer Informationen in der Vorgangsübersicht des Antragstellers durch das Portal

Schritt 6: Versand einer Benachrichtigung über Anmeldung zur Nutzung der MDS oder Änderung der hinterlegten Informationen an technischen Ansprechpartner

Schritt 7: Abholung der Zugangsdaten zur Massendatenschnittstelle aus der Vorgangsübersicht durch den Antragsteller/Portalnutzer

3.2. Aktualisierungen

Zur Aktualisierung der Informationen, welche bei der Freischaltung zur Nutzung der Massendatenschnittstelle angegeben wurden, kann das oben genannte Antragsformular mit den geänderten Angaben erneut übermittelt werden. Dabei ist zu beachten, dass das Änderungsformular vollständig auszufüllen ist, auch Angaben, die sich nicht ändern.

Hinweis

Bei einer Aktualisierung ab dem 01.01.2025 wird nur noch Version 2 unterstützt.



3.3. Deaktivierung

In der aktuellen Version der DIP-Schnittstelle ist die Deaktivierung des Zugangs möglich.

4. Nutzung DIP über das Portal

Dieses Kapitel beschreibt den Upload einer Massendatenlieferung als XML-Datei über das BZSt online.portal. Es ist geplant, diese Funktion bis Mitte 2025 zur Verfügung zu stellen. Nähere Einzelheiten und Erläuterungen werden in einer späteren Version des Kommunikationshandbuchs veröffentlicht.

Teil II Datenübertragung

Die nachfolgenden Kapitel beschreiben die weitere Vorgehensweise bei der Datenübertragung für Verfahren, die dem DIP-Standard unterliegen (siehe Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**) Für die Datenübermittlung bei Fachverfahren, die bisher nicht dem DIP-Standard unterliegen, sind die jeweils fachspezifischen Dokumentationen zu beachten.

5. Anforderungen an die zu übertragenden Dateien

5.1. Aufbau des DIP XML-Schemas Version 1

5.1.1. Adresse zum DIP XML-Schema

Das DIP XML-Schema wird Ihnen auf der Homepage des BZSt unter „Massendaten“ zum Download angeboten.

5.1.2. Aufbau des Wurzelements DIP

Es wurde eine verfahrensübergreifende, standardisierte XML-Schema-Definition für die Nutzung der Massendatenschnittstelle im POST 2.0 geschaffen, in die die jeweiligen Fachverfahrensdaten im fachverfahrensspezifischen XML einzubetten sind.

Das Wurzelement *dip* ist vom ComplexType *DipType*. DIP steht dabei für Digitaler Posteingang. Das Wurzelement *dip* wird wie ein „Umschlag“ um die eigentlichen Dateninhalte herumgelegt und besteht aus einem *header*- und einem *body*-Teil. Letzterer enthält die verfahrensspezifischen Nutzdaten. Im Attribut *version* wird die Version des verwendeten Dip-XML-Schemas hinterlegt, um bei Anpassungen des Schemas Nachrichten verschiedener Schema-Versionen unterscheiden zu können. Die aktuelle Version ist 1.0.

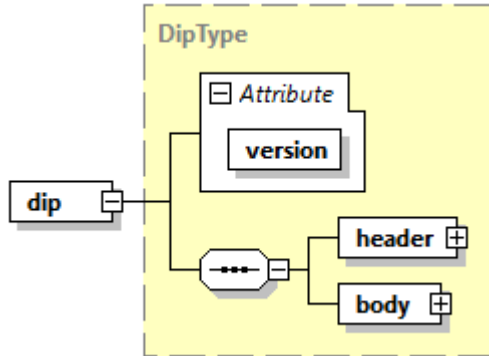


Abbildung 4 Element *dip* (DipType)

5.1.3. Aufbau des Headers

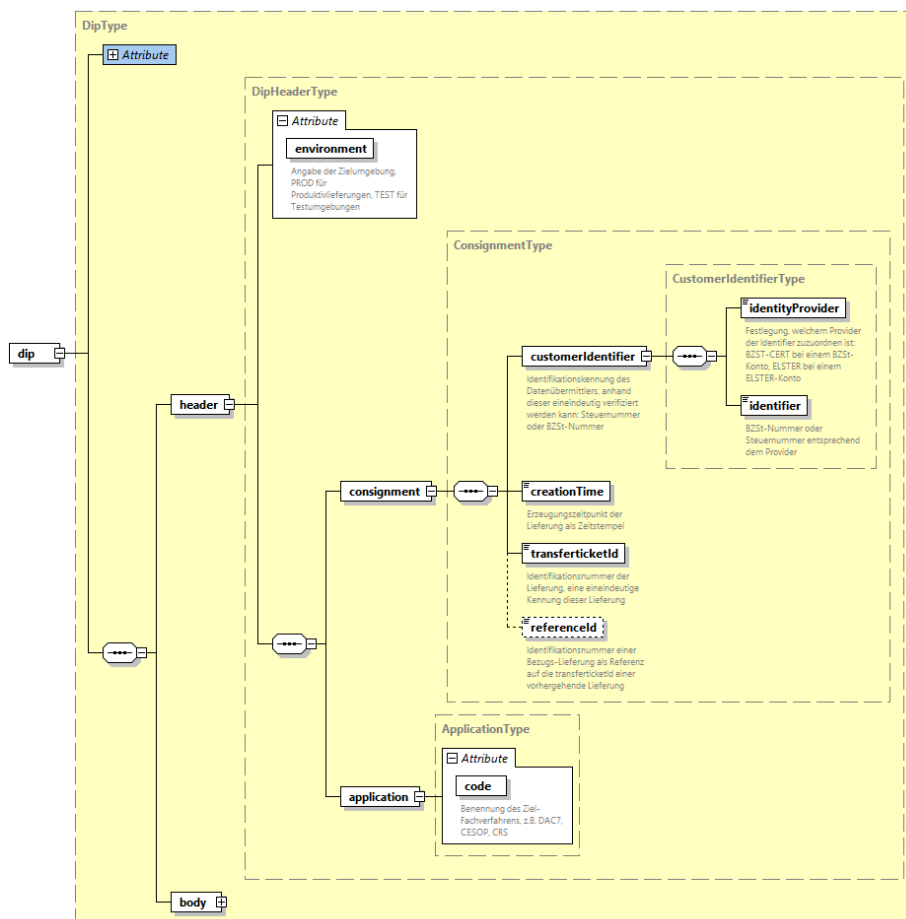


Abbildung 5 Element header (DipHeaderType)

Alle Attribute im *dip*-Header wie auch im *dip*-Body sind vom Datenübermittler auszufüllen, insbesondere die Pflichtangaben.

Element	Beschreibung	Bemerkungen
---------	--------------	-------------



Kommunikationshandbuch DIP-Standard

Standardisierte Datenübermittlung an das BZSt über die
Massendatenschnittstelle DIP

header/@environment	Angabe der Zielumgebung, PROD für Produktivlieferungen, TEST für Testumgebungen.	Angabe ist verpflichtend.
application/@code	Angabe, für welches Fachverfahren die Lieferung bereitgestellt wird.	Der zu verwendende Code wird durch das jeweilige Fachverfahren kommuniziert.
consignment/creationTime	Erzeugungszeitpunkt der Lieferung als Zeitstempel.	
consignment/customerIdentifier	Identifikationskennung des Datenübersmitters, anhand dieser eindeutig verifiziert werden kann: Steuernummer oder BZSt-Nummer (s.a. Kapitel 3 Freischaltung, Aktualisierung und Deaktivierung der DIP Schnittstelle und 3.3 Deaktivierung)	
consignment/customerIdentifier/identityProvider	Festlegung, welchem Provider die Identifikationskennung (Ordnungsmerkmal) zuzuordnen ist: <i>BZST-CERT</i> aufgrund der Freischaltung über ein BZSt-Konto, <i>ELSTER</i> aufgrund der Freischaltung über ein ELSTER-Konto.	mögliche Wert: <i>BZST-CERT</i> , <i>ELSTER</i>
consignment/customerIdentifier/identifier	Ordnungsmerkmal beim angegebenen Provider. Als Ordnungsmerkmal werden <ul style="list-style-type: none">• die Steuernummer (bei der Identifizierung im Portal über Elster) oder• die BZST-Nummer sowie der zugehörige Typ (ELSTER oder BZST-CERT) verwendet.	
consignment/transferticketId	Identifikationsnummer der Lieferung, eine eindeutige Kennung dieser Lieferung.	
consignment/referenceId	Identifikationsnummer einer Bezugs-Lieferung als Referenz auf die transferticketId einer vorhergehenden Lieferung.	Angabe ist optional.

Tabelle 6 Beschreibung der Elemente im DipHeader



5.1.4. Aufbau der verfahrensspezifischen Nutzdaten

Der Aufbau des Nutzdatenschemas ist der Schnittstellenbeschreibung des jeweiligen Fachverfahrens und den durch das Fachverfahren bereitgestellten XML-Schemata zu entnehmen.

Im *body*-Zweig können mehrere fachliche Datensätze als *consignmentItem* im XML positioniert werden. Die Positionen in der übergebenen Liste werden bei der Rückgabe des Protokolls in dem Feld *consignmentItemPosition* referenziert. Unter dem Element *data* befindet sich das fachverfahrensspezifische XML. Im DIP-XML-Schema wird dies wegen des generischen Ansatzes von DIP durch den Datentyp *anyType* dargestellt. Das Element *bopAccountId* wird für Rücklieferungen vom Fachverfahren (z.B. Verwaltungsakte) an das individuelle Konto des Antragstellers im BZSt online.Portal (POST 2.0) verwendet.

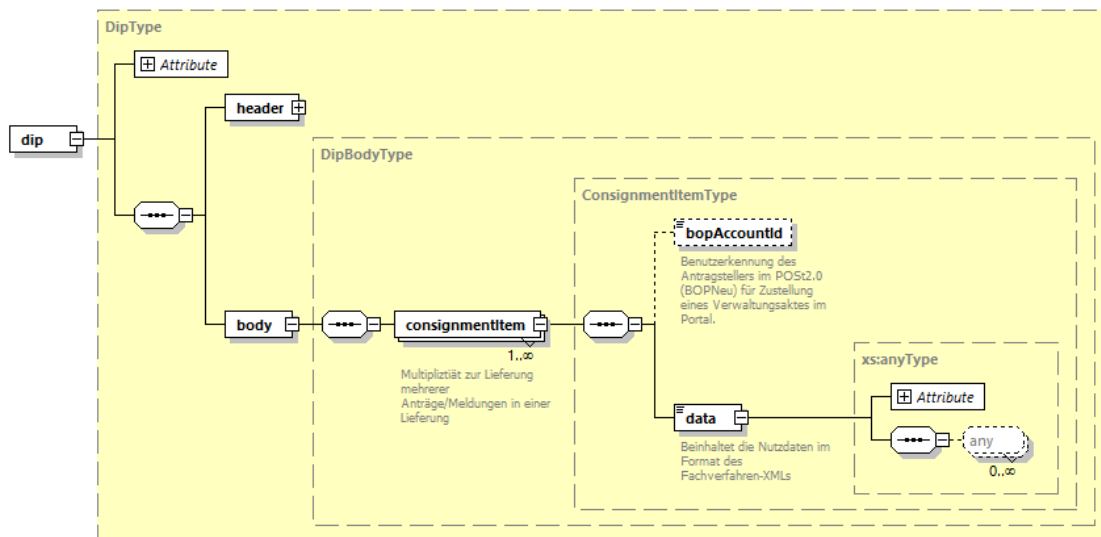


Abbildung 6 Element body (DipBodyType)

Element	Beschreibung	Bemerkungen
consignmentItem/bopAccountId	Angabe einer Benutzerkennung zur Zustellung von Bescheiden, Rückfragen etc. im BOP.	Angabe ist optional.
consignmentItem/data	Datenfeld, in dem die Nutzdaten entsprechend des jeweiligen Fachverfahren-XMLs abgelegt werden.	

Tabelle 7 Beschreibung der Elemente im DipBody

5.1.5. Beispiel

Im folgenden Beispiel wird eine Massendatenlieferung skizziert, die als Nutzdaten ein *consignmentItem* für das Fachverfahren DAC7 besitzt. Der fachliche Bereich des XML ist nur angerissen (visualisiert durch "...").



Beispiel DIP-XML einer Massendatenlieferung

```
<dip version="1.0" xmlns="http://itzbund.de/ozg/bzst/post/dip/v1/">
  <header environment="TEST">
    <consignment>
      <customerIdentifier>
        <identityProvider>BZST-CERT</identityProvider>
        <identifier>BZ12345</identifier>
      </customerIdentifier>
      <creationTime>2023-09-01T04:55:22Z</creationTime>
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>
    </consignment>
    <application code="DAC7"/>
  </header>
  <body>
    <consignmentItem>
      <data>
        <dpi:DPI_OECD version="1.0" xmlns:dpi="urn:oe.cd:ties:dpi:v1">
          <dpi:MessageSpec>
            <dpi:SendingEntityIN>Steuernummer</dpi:SendingEntityIN>
            <dpi:TransmittingCountry>DE</dpi:TransmittingCountry>
            ...
          </dpi:MessageSpec>
          <dpi:DPIBody>
            <dpi:PlatformOperator>
              <dpi:ResCountryCode>DE</dpi:ResCountryCode>
            ...
          </dpi:DPI_OECD>
        </data>
      </consignmentItem>
    </body>
  </dip>
```

Auch die Vergabe eine Namespaces ist möglich

```
<dip:dip xmlns:dip="http://itzbund.de/ozg/bzst/post/dip/v1/" version="1.0">
  <dip:header environment="TEST">
    ...
```

5.2. Aufbau des DIP XML-Schemas Version 2

5.2.1. Adresse zum XML-Schema

Das DIP XML-Schema für die Version 2 der DIP-Schnittstelle wird Ihnen auf der Homepage des BZSt unter „Massendaten“ zum Download angeboten.



5.2.2. Aufbau des Wurzelements DIP

Es wurde eine verfahrensübergreifende, standardisierte XML-Schema-Definition für die Nutzung der Massendatenschnittstelle im POST 2.0 geschaffen, in die die jeweiligen Fachverfahrensdaten im fachverfahrensspezifischen XML einzubetten sind.

Das Wurzelement dip wird wie ein „Umschlag“ um die eigentlichen Dateninhalte herumgelegt und besteht aus einem *header*- und einem *body*-Teil. Letzterer enthält die verfahrensspezifischen Nutzdaten. Im Attribut *version* wird die Version des verwendeten Dip-XML-Schemas hinterlegt, um bei Anpassungen des Schemas Nachrichten verschiedener Schema-Versionen unterscheiden zu können. Die aktuelle Version für die Schnittstelle in Version 2 ist 2.0.

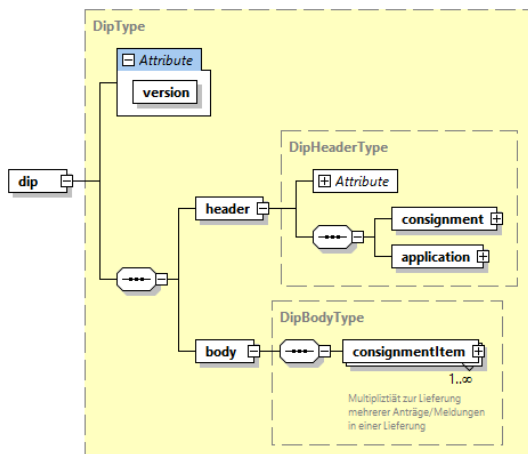


Abbildung 7 Element dip (DipType)



5.2.3. Aufbau des Headers

Der Header ist in folgender Abbildung dargestellt.

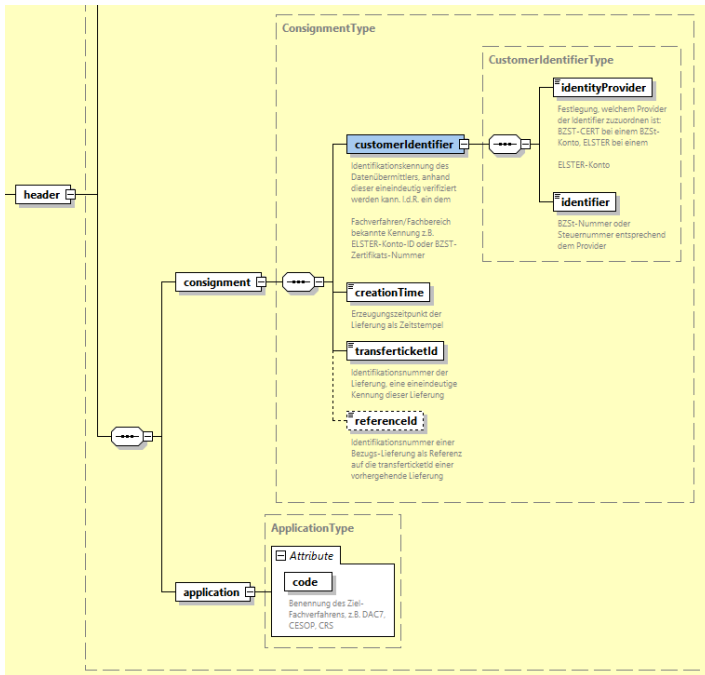


Abbildung 8 Element header (DipHeaderType)

Alle Attribute im *dip*-Header sind vom Datenübermittler auszufüllen. Die Pflichtfelder ergeben sich aus der XSD.

Element	Beschreibung	Bemerkungen
header/@environment	Angabe der Zielumgebung	PROD für Produktivlieferungen, TEST für Testumgebungen
application/@code	Angabe, für welches Fachverfahren die Lieferung bereitgestellt wird.	Der zu verwendende Code wird durch das jeweilige Fachverfahren kommuniziert.
consignment/creationTime	Erzeugungszeitpunkt der Lieferung als Zeitstempel.	
consignment/customerIdentifier	Identifikationskennung des Datenübermittlers, anhand dieser	



	eindeutig verifiziert werden kann: Steuernummer oder BZSt-Nummer (s.a. Kapitel 3 Freischaltung, Aktualisierung und Deaktivierung der DIP Schnittstelle und 3.3 Deaktivierung)	
consignment/customer Identifier/identityProv ider	Festlegung, welchem Provider die Identifier- Kennung (Ordnungsmerkmal) zuzuordnen ist: <i>BZST-CERT</i> aufgrund der Freischaltung über ein BZSt-Konto, <i>ELSTER</i> aufgrund der Freischaltung über ein ELSTER-Konto.	mögliche Wert: <i>BZST- CERT, ELSTER</i>
consignment/customer Identifier/identifier	Ordnungsmerkmal beim angegebenen Provider. Als Ordnungsmerkmal werden <ul style="list-style-type: none"> • die Steuernummer (bei der Identifizierung im Portal über Elster) oder • die BZST-Nummer sowie der zugehörige Typ (ELSTER oder BZST-CERT) verwendet.	
consignment/transferticketId	Identifikationsnummer der Lieferung, eine eindeutige Kennung dieser Lieferung.	
consignment/reference Id	Identifikationsnummer einer Bezugs- Lieferung als Referenz auf die transferticketId einer vorhergehenden Lieferung.	Angabe ist optional.

Tabelle 8 Beschreibung der Elemente im DipHeader

5.2.4. Aufbau der verfahrensspezifischen Nutzdaten

Der Aufbau des Nutzdatenschemas ist der Schnittstellenbeschreibung des jeweiligen **Fachverfahrens** und den durch das Fachverfahren bereitgestellten XML-Schemata zu entnehmen.

Im *body*-Zweig können mehrere fachliche Datensätze als *consignmentItem* im XML positioniert werden. Die Positionen in der übergebenen Liste werden bei der Rückgabe des Protokolls in dem Feld *consignmentItemPosition* referenziert. Unter dem Element *data* befindet sich das fachverfahrensspezifische XML. Im DIP-XML-Schema wird dies wegen des generischen Ansatzes von DIP durch den Datentyp *anyType* dargestellt. Das Element *bopAccountId* wird für Rücklieferungen vom Fachverfahren (z.B. Verwaltungsakte) an das individuelle Konto des Antragstellers im BZSt online.portal (POSt 2.0) verwendet.

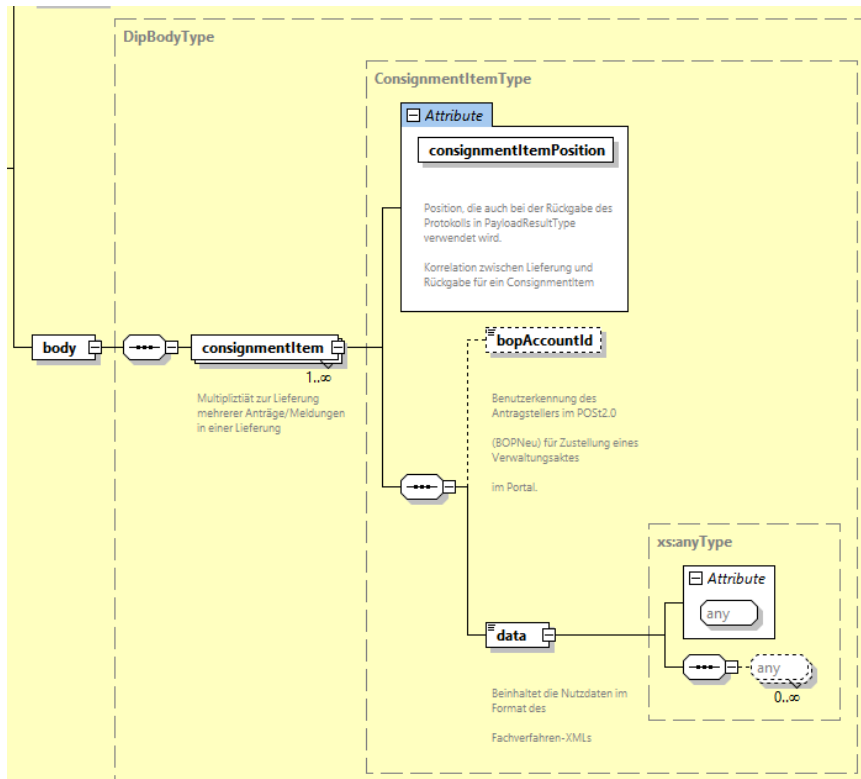


Abbildung 9 Element body (DipBodyType)

Element	Beschreibung	Bemerkungen
consignmentItem /consignmentItemPosition	Das Pflichtattribut <i>consignmentItemPosition</i> identifiziert das Item. Diese Id wird innerhalb des Protokolls verwendet, um das jeweilige Ergebnis dem Item zuzuordnen.	
consignmentItem /bopAccountId	Angabe einer Benutzerkennung zur Zustellung von Bescheiden, Rückfragen etc. im BOP.	Angabe ist optional.
consignmentItem /data	Datenfeld, in dem die Nutzdaten entsprechend des jeweiligen Fachverfahren-XMLs abgelegt werden.	

Tabelle 9 Beschreibung der Elemente im DipBody

5.2.5. Beispiel

Im folgenden Beispiel wird eine Massendatenlieferung skizziert, die als Nutzdaten ein **consignmentItem** mit der Position 42 für das Fachverfahren CESOP besitzt. Der fachliche Bereich des XML ist verkürzt dargestellt (visualisiert durch "...").

Beispiel DIP-XML einer Massendatenlieferung

```
<dip xmlns="http://itzbund.de/ozg/bzst/post/dip/v2/" version="2.0">
  <header environment="TEST">
    <consignment>
```



```
<customerIdentifier>
  <identityProvider>BZST-CERT</identityProvider>
  <identifier>BZ12345</identifier>
</customerIdentifier>
<creationTime>2023-09-01T04:55:22Z</creationTime>
<transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>
</consignment>
<application code="CESOP"/>
</header>
<body>
  <consignmentItem consignmentItemPosition="42">
    <data>
      <cesop:CESOP xmlns:cesop="urn:ec.europa.eu:taxud:fiscalis:cesop:v1" version="4.0">
        ....
      </cesop:CESOP>
    </data>
  </consignmentItem>
</body>
</dip>
```

Auch die Vergabe eines Namespaces wird von der Schnittstelle unterstützt.

```
<dip:dip xmlns:dip="http://itzbund.de/ozg/bzst/post/dip/v2/" version="2.0">
  <dip:header environment="TEST">
    ...
```

5.3. Weitere Anforderungen

5.3.1. Vergabe der Transferticket-ID

Bei der Transferticket-ID handelt es sich um eine eindeutige Zeichenfolge zur Identifizierung einer Lieferung. Diese muss bei der initialen Lieferung durch Sie generiert und unter XPath `dip/header/consignment/transferticketId` abgelegt werden. Die Eindeutigkeit muss über all Ihre Lieferungen hinweg gegeben sein. Geeignet wäre hier beispielsweise eine UUID.

Die vergebene Transferticket-ID kann später als Reference-ID in einer Rückantwort oder Nachlieferung zur ursprünglichen Lieferung referenziert werden.

5.3.2. XSD-Schema Validierung

Die XML-Datei wird gegen das aktuelle DIP-XML-Schema validiert. Zusätzlich werden die Nutzdaten unter XPath `dip/body/consignmentItem/data` gegen das entsprechende XSD-Schema des jeweiligen Fachverfahrens validiert werden.



5.3.3. Encoding in UTF-8

Für die Erstellung der XML-Datei ist als Encoding UTF-8 zu verwenden. Abweichende Encodings werden nicht unterstützt und können zu einer Abweisung führen. Es wird empfohlen, nur die üblicherweise in Namens- und Adressdaten enthaltenen Zeichen der UTF-8 Kodierung zu nutzen. Nur so sind eine unverfälschte Speicherung und Verarbeitung der Daten gewährleistet. Als Orientierungshilfe kann der von der Koordinierungsstelle für IT-Standards (KoSIT) des IT-Planungsrats im Bundesministerium des Innern definierte Satz der lateinischen Zeichen herangezogen werden.

5.3.4. Unzulässige Zeichen und Beschränkungen

DIP nimmt Daten gemäß UTF-8 grundsätzlich ohne Einschränkung an. In den einzelnen Fachverfahren können jedoch unzulässige Zeichen und weitere Restriktionen festgelegt sein, die zu einer Abweisung führen können.

5.3.5. Dateigrößenbeschränkung

Die zu übermittelnde XML-Datei sollte kleiner als 1 GiB (Gibibyte = 1024^3 . 1 GB = Gigabyte = 1000^3) sein. Eine Überschreitung dieser Maximalgröße führt zu einer Abweisung. Für die Anhangsdatei gilt dieselbe Maximalgröße.

6. Übertragung der Dateien

6.1. Aufbau der Verbindung

Folgende Voraussetzungen müssen erfüllt sein, um Massendaten über die Schnittstelle übertragen zu können:

- Der Kunde ist für die Nutzung der DIP-Schnittstelle freigeschaltet
 - und hat somit eine DIP-ID bekommen
 - und hat den privaten Schlüssel zur Verfügung, auf dessen Basis das Zertifikat erstellt wurde, das bei der Freischaltung übergeben wurde.
- Vor jeder Kommunikation mit der DIP-Schnittstelle sollte ein Access-Token abgerufen werden.

Weitere Informationen können dem Kapitel [2 Technische Beschreibung der Kommunikation](#) entnommen werden.

6.2. REST-Schnittstelle der DIP-Schnittstelle aufrufen

Folgende URL-Präfixe kommen in den zwei Versionen der Schnittstelle zum Einsatz:



Zweck	Version	Präfix	Anmerkung
Autorisierung	1 und 2	<Domäne des Systems>/auth/	
Datenlieferung	1	<Domäne des Systems>/dip/start <Domäne des Systems>/dip/md/...	Der Endpunkt für „start“ liegt direkt unter „dip“ Alle anderen Endpunkte liegen unter „/dip/md“
Datenlieferung	2	<Domäne des Systems>/dip/v2/md/...	Alle Endpunkte liegen unter „/dip/v2/md“

Tabelle 10 URL-Präfixe in den zwei Versionen der Schnittstelle

6.2.1. Prüfung des Access-Tokens

Die Prüfung des Access-Tokens erfolgt bei jedem Aufruf der DIP-Schnittstelle. Die Reihenfolge der für eine Datenlieferung aufzurufenden Dienste ist nachfolgend beschrieben. In jedem dieser Aufrufe ist das Access-Token in den Header zu integrieren, welches in Kapitel 2 [Technische Beschreibung der Kommunikation](#) beschrieben ist.

REST-Aufruf der Massendatenschnittstelle mit Token im Produktivsystem

```
GET https://mds.bzst.bund.de/dip/...  
Authorization: bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2kiOiA6IChxQ...
```

Ein beispielhafter Ablauf einer vollständigen Datenlieferung an die DIP-Schnittstelle findet sich in Kapitel 6.5.1 [Möglicher Ablauf einer Datenlieferung im Produktivsystem](#).

Das Access-Token wird bei jedem Aufruf durch die DIP-Schnittstelle geprüft und damit sichergestellt, dass nur berechtigte Aufrufe erfolgen.



REST-Zugriff mit Access-Token

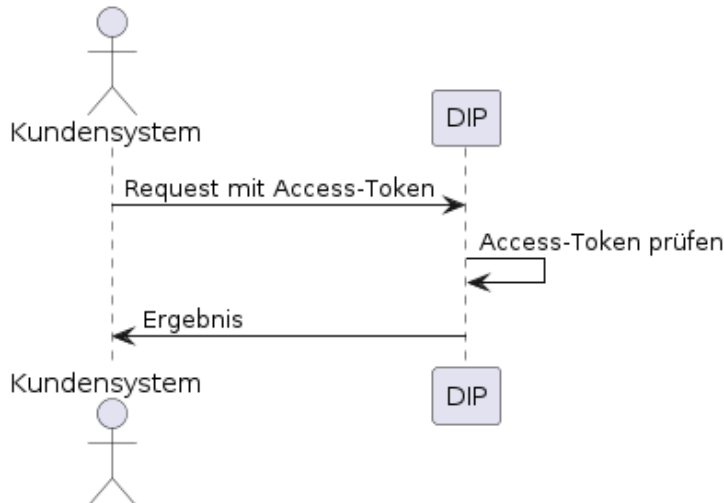


Abbildung 10 REST-Zugriff mit Access-Token

Empfehlung für die Nutzung der DIP-Schnittstelle

Die Gültigkeitsdauer des Access-Tokens ist aus Sicherheitsgründen auf wenige Sekunden beschränkt. Daher sollte vor jedem Aufruf der REST-Schnittstelle ein neues Access-Token erfragt werden. Natürlich kann in der Software des Kunden auch die Gültigkeit des Tokens geprüft werden und das Token bis zum Ablauf der Gültigkeit verwendet werden.

Die Anfrage nach dem Access-Token kann folgende http Response Status Codes liefern:

Code	Status	Bedeutung
200	OK	Einreichung erfolgreich
400	Bad Request (z.B. invalid_client)	Etwas stimmt mit dem Request nicht. Beispiele: veraltete Token, Startzugangstoken wird mehr als einmal verwendet, falsche URL, falsche Request-Parameter oder falscher Algorithmus für die Signierung des JWT
401	Unauthorized	Tritt bei einem ungültigen Token oder bei einer nicht gültigen Key und Client ID Kombination auf.
403	Forbidden (z.B. unbekannte URL)	Der Pfad der URL ist nicht zulässig
404	Not Found	Zielverfahren nicht gefunden
405	Method Not Allowed	Eine nicht unterstützte HTTP Methode wird benutzt.



410	Gone	Datentransfer wurde bereits beendet
424	Failed Dependency	Der Datenübermittler ist dem Vorgang nicht zugeordnet
429	Too many Requests	Wenn mehr als die maximalen Requests je Minute eingereicht werden
500	Internal Server Error	Internal Fehler in der MDS Verarbeitung.
502	Bad Gateway	Verbindung konnte nicht aufgebaut werden z.B. wegen Netzwerk-Störungen.
503	Service Unavailable	Service ist nicht erreichbar.

Tabelle 11 HTTP Response Status Codes

6.3. Ablauf der Datenübertragung

Eine Massendaten-Lieferung besteht aus der signierten XML-Datei und ggf. Anhängen oder Nachweisen in einer ZIP-Datei. Als Vorgehensweise zur Signierung der zu sendenden XML-Datei wird die **Enveloping-Signatur** für Daten der Version 1 und **Enveloped-Signatur** für Daten der Version 2 unterstützt. Hierbei ist die Signatur in der zu übertragenden XML-Datei integriert, indem die Signatur als Wrapper um das Daten-XML herumgelegt wird (vgl. Beispiel in Kapitel 2 [Technische Beschreibung der Kommunikation](#)).

Wird die signierte XML-Datei oder der Anhang mehrfach hochgeladen, werden die Dateien als **Update** behandelt. Dabei wird die bisherige Datei gelöscht und die neue gespeichert.

■ Hinweis zur Anzahl erlaubter Datenlieferungen

Die Massendatenschnittstelle nimmt maximal **10 Datenlieferungen pro Minute** von einem Datenübermittler entgegen. Dieses Limit bezieht sich auf den Endpunkt "Einreichung starten" (*dip/start*). Wird dieses Limit überschritten, wird der Fehlercode 429 zurückgegeben. Nach einer entsprechenden Wartezeit können nach dieser Meldung wieder Meldungen angemeldet werden.

Ein Upload besteht aus den folgenden Schritten bzw. REST-Aufrufen, die automatisiert in eine Batchverarbeitung eingebunden werden können.

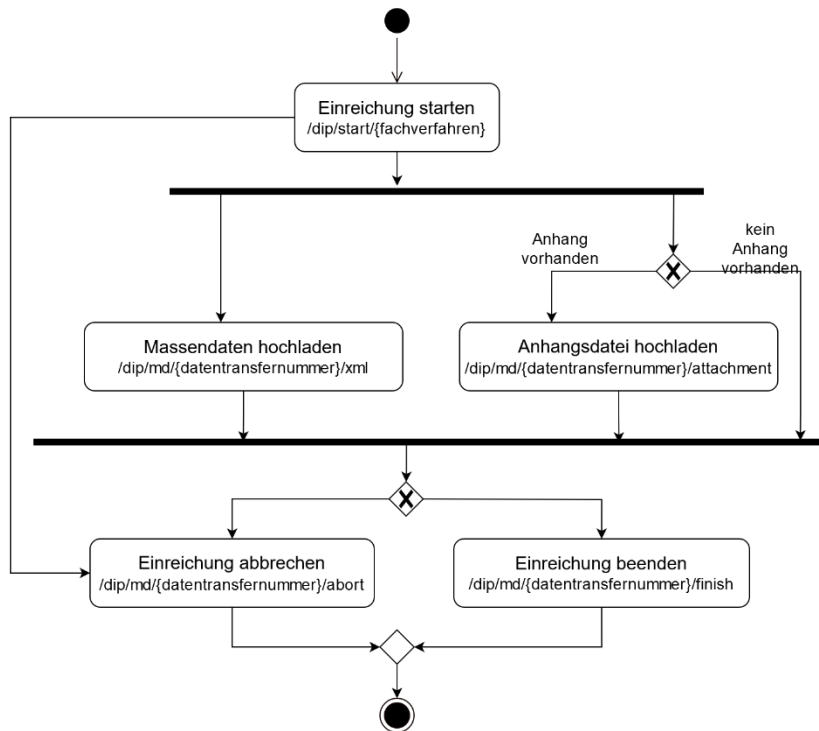


Abbildung 11 Ablauf einer Datenübermittlung

Schritt 1: Der Aufruf von "Einreichung anmelden" unter Angabe des Fachverfahrens, liefert eine Datentransfernummer, die eine Klammer um die zu liefernden Daten bildet und bei jedem Aufruf angegeben wird.

Schritt 2: Datenübertragung unter Angabe der Datentransfernummer

- ➔ Daten-XML übermitteln
- ➔ optional: Anhang übermitteln – die Anhänge in der ZIP Datei beziehen sich auf die gesamte Lieferung und nicht auf einzelne ConsignmentItems.
- ➔ Bis zu einem "/finish" (Einreichung Beenden) werden erneut hochgeladene Dateien akzeptiert und als Update zu den alten betrachtet. Die alten Dateien werden überschrieben.

Schritt 3: Einreichung unter Angabe der Datentransfernummer beenden

Schritt 4: Abruf des Verarbeitungsprotokolls unter Angabe der Datentransfernummer

Bei der Datenübertragung ist **keine** festgelegte Reihenfolge zur Übermittlung der Dateien zu berücksichtigen. Nach der Anmeldung können die Dateien in beliebiger Reihenfolge gesendet werden.

Die Einreichung kann zu jedem Zeitpunkt nach der Anmeldung alternativ zur Beendigung unter Angabe der Datentransfernummer auch abgebrochen werden.



Es ist erforderlich, dass ein Abbrechen oder ein Beenden durch den Kunden signalisiert und damit die Datenlieferung beendet wird. Erfolgt dies nicht innerhalb von **24 Stunden** nach der Einreichungsanmeldung, so wird der Einlieferungsprozess abgebrochen und ein Fehlerprotokoll erstellt.

6.4. Abruf des Ergebnisprotokolls

Das Ergebnisprotokoll steht i.A. **nicht** sofort nach der Einreichung bereit. Im Kommunikationshandbuch des Fachbereichs finden sich Informationen zu den zu erwartenden Bearbeitungszeiten. Der Abruf des Ergebnisprotokolls muss regelmäßig durchgeführt werden, bis er erfolgreich war. Das Ergebnisprotokoll steht nach dessen Erstellung **3 Monate** zum Abruf bereit. Bei Bereitstellung des Ergebnisprotokolls wird eine E-Mail an den angegebenen technischen Ansprechpartner gesendet, die über die Bereitstellung informiert.

Der Abruf der Protokolle von Datenlieferungen gliedert sich in folgende Schritte.

Die Umsetzung durch Aufrufe der REST Schnittstelle ist im Abschnitt "Möglicher Ablauf einer Datenlieferung im Produktivsystem" dargestellt.

Schritt 1: Abruf aller Transferrnummern, für die die Abholung nicht bestätigt wurde

Schritt 2: Abruf der Protokolle zu den Protokollnummern

Schritt 3: Bestätigung des Protokollabrufs

Bis zur Bestätigung des Protokollabrufs für eine Transferrnummer wird die Transferrnummer beim Abruf aller Transferrnummern zurückgegeben.

⚠ Ein Abruf von Protokollen für Transferrnummern, die **nicht** zuvor mit dem "Abruf aller Transferrnummern" zurückgegeben wurden, ist zu vermeiden.

6.5. REST-API der Massendatenschnittstelle

6.5.1. Möglicher Ablauf einer Datenlieferung im Produktivsystem

Folgende Requests stellen einen möglichen Ablauf bei der **Einreichung** von Massendaten an DIP dar:

Request zum Einleiten einer Massendatenlieferung - "Einreichung starten":

- **POST**
<https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token> - Startzugangstoken (Das Token, das mit dem Private Key/Zertifikat vom Kunden generiert ist)
⇒ Ergebnis: Statuscode 200 mit dem Access-Token



- **POST**
https://mds.bzst.bund.de/dip/start/{fachverfahren} - unter Verwendung des ersten Tokens
⇒ Ergebnis: Statuscode 201 und die Datentransfernummer

Übertragen der Daten unter Angabe der erhaltenen Datentransfernummer – „Massendaten hochladen“

- **POST**
https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
⇒ Ergebnis: Statuscode 200 mit dem Access-Token
- **PUT**
https://mds.bzst.bund.de/dip/md/{datentransfernummer}/xml - unter Verwendung des neuen Tokens
⇒ Ergebnis: Statuscode 200

Übertragen der Anhänge unter Angabe der erhaltenen Datentransfernummer – „Anhangsdatei hochladen“

- **POST**
https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
⇒ Ergebnis: Statuscode 200 mit dem Access-Token
- **PUT**
https://mds.bzst.bund.de/dip/md/{datentransfernummer}/attachment - unter Verwendung des neuen Tokens
⇒ Ergebnis: Statuscode 200

Request zum Erklären, dass die Lieferung vollständig übertragen wurde – „Einreichung beenden“

- **POST**
https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
⇒ Ergebnis: Statuscode 200 mit dem Access-Token
- **PATCH**
https://mds.bzst.bund.de/dip/md/{datentransfernummer}/finish - unter Verwendung des neuen Tokens
⇒ Ergebnis: Statuscode 200

Wenn ein **Abbruch** der Lieferung notwendig werden sollte, ist folgender Request zu senden:

Request zum Erklären, dass die Lieferung abgebrochen und gelöscht werden soll – „Einreichung abbrechen“

- **POST**



https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
⇒ Ergebnis: Statuscode 200 mit dem Access-Token

- **PATCH**

https://mds.bzst.bund.de/dip/md/{datentransfernummer}/abort - unter
Verwendung des neuen Tokens
⇒ Ergebnis: Statuscode 200

Nachdem die Daten übermittelt wurden, kann regelmäßig geprüft werden, ob die Verarbeitung der Daten beendet wurde und ein Protokoll vorliegt. Hierfür werden alle Transfernummern, für die ein Protokoll vorliegt, das noch nicht abgerufen wurde, erfragt. Für diese Transfernummern wird das **Protokoll abgerufen** und der Empfang bestätigt.

Abruf des Ergebnisprotokolls

- **POST**

https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
⇒ Ergebnis: Statuscode 200 mit dem Access-Token

- **GET**

https://mds.bzst.bund.de/dip/md/protocolnumbers
⇒ Ergebnis: Statuscode 200 und XML-Liste mit Datentransfernummern, zu denen ein Protokoll vorliegt
⇒ Ist die Liste **leer**, sollte der Request zu einem späteren Zeitpunkt wiederholt werden
⇒ Ist die Liste **nicht leer**, kann der folgende Request mit der enthaltenen Datentransfernummer gestartet werden (vorher ist wieder ein neues Access-Token zu erfragen)

Für alle erhaltenen Transfernummern:

- **POST**

https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token
⇒ Ergebnis Statuscode 200 mit dem Access-Token

- **GET**

https://mds.bzst.bund.de/dip/md/{datentransfernummer}/protocol - unter
Verwendung des neuen Tokens und einer korrekten Datentransfernummer
⇒ Statuscode 200: DipProtocol

- **POST**

https://mds.bzst.bund.de/auth/realms/mds/protocol/openid-connect/token



⇒ Ergebnis Statuscode 200 mit dem Access-Token

- **PATCH**

https://mds.bzst.bund.de/dip/md/{datentransfernummer}/protocol - Bestätigung des Abrufs unter Verwendung des neuen Tokens und einer korrekten Datentransfernummer

⇒ Statuscode 200

6.5.2. OpenAPI-Dokumentation

Die Spezifikation wird Ihnen auf der Homepage des BZSt unter „Massendaten“ zum Download angeboten und enthält die folgenden Dienste:

- **POST**

/auth/realms/{realm}/protocol/openid-connect/token (*Endpunkt, aber nicht Teil der OpenAPI)

⇒ Neues Access Token

- **POST**

/dip/start/{fachverfahren}

⇒ Start einer Massendatenlieferung

- **PUT**

/dip/md/{datentransfernummer}/xml

⇒ Daten-XML übermitteln

- **PUT**

/dip/md/{datentransfernummer}/attachment

⇒ Anhang übermitteln

- **PATCH**

/dip/md/{datentransfernummer}/finish

⇒ Finalisieren einer Massendatenlieferung

- **PATCH**

/dip/md/{datentransfernummer}/abort

⇒ Abbrechen einer Massendatenlieferung

- **GET**

/dip/md/{datentransfernummer}/protocol

⇒ Abruf des Verarbeitungsprotokolls

- **PATCH**

/dip/md/{datentransfernummer}/protocol

⇒ Erhalt des Protokolls bestätigen



- **GET**
`/dip/md/protocolnumbers`
⇒ Abruf Datentransfernummern mit verfügbarem Protokoll

6.5.2.1. Detailbeschreibung der OpenAPI

Generierung eines neuen Access Tokens

POST `/auth/realms/{mds|ktst}/protocol/openid-connect/token`

Ein Access-Token für die Übermittlung von Massendaten anfragen.

Parameters

Name	Description
grant_type	client_credentials
client_assertion_type	urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
client_assertion	Selbst generierte Zugangstoken

Header

Name	Description
Content-Type	application/x-www-form-urlencoded

Request Body

Type	Content
-	-

Response

Status Code	Description	Response Body (Beispiele)
200	Einreichung erfolgreich	{ "access_token": <base64 Token> "expires_in": 300, "refresh_expires_in": 0, "token_type": "Bearer", "not-before-policy": 0, "scope": "email profile"}
400	Bad Request	Startzugangstoken wurde mehrfach verwendet { "error": "invalid_client", "error_description": "Client authentication with signed JWT failed: Token reuse detected"}



Kommunikationshandbuch DIP-Standard

Standardisierte Datenübermittlung an das BZSt über die
Massendatenschnittstelle DIP

		<p>Das Startzugangstoken ist nicht mehr gültig</p> <pre>{ "error": "invalid_client", "error_description": "Client authentication with signed JWT failed: Token is not active"}</pre> <p>Falschen Request Parameters / fehlendes Token</p> <pre>{ "error": "invalid_client", "error_description": "Invalid client credentials"}</pre> <p>Falsche JWT Signature ALG:</p> <pre>{"error": "invalid_client", "error_description": "Client authentication with signed JWT failed: Signature on JWT token failed validation"}</pre> <pre>{ "error": "invalid_client", "error_description": "Unable to load public key" }</pre>
401	Unauthorized	Eine ungültige Autorisierung kann auftreten, wenn eine falsche Client-Id und falsche Keys benutzen werden. Ebenso, wenn das Access Token ist nicht mehr gültig ist.
403	Forbidden	Der in der URL verwendete Pfad ist nicht bekannt oder gesperrt.
405	Method not allowed	<pre>{ "error": "RESTEASY003650: No resource method found for GET/PUT/DELETE, return 405 with Allow header" }</pre>
503	Service Unavailable	

6.5.2.2. Start einer Massendatenlieferung

POST /dip/start/{fachverfahren}

Anfordern einer Datentransfernummer zur Identifikation der Lieferung und Einleiten einer Massendatenlieferung

Parameters



Name	Description
fachverfahren	Das Ziel-Fachverfahren

Werte für das Fachverfahren werden in den Kommunikationshandbüchern der jeweiligen Fachverfahren veröffentlicht (vgl. Kapitel 5 [Anforderungen an die zu übertragenden Dateien](#), unter dem Abschnitt "Aufbau des Elements DipHeaderTyp", im Dip-XML unter *dip/header/application/@code*).

Header

Name	Description
Authorization	JWT Access Token

Responses

Status Code	Description	Content
201	Datentransfer erfolgreich initiiert	TEXT/PLAIN Datentransfernummer
401	Ungültiger Authorization Header	-
404	Zielfachverfahren nicht gefunden	-
429	Der Client hat zu viele Anfragen in der letzten Minute gesendet	-
500	Interner Fehler	-

6.5.2.3. Daten-XML übermitteln

PUT /dip/md/{datentransfernummer}/xml

Einreichung der Daten-XML unter Angabe der Datentransfernummer der Massendatenlieferung. Existiert bereits eine XML-Datei zu der Datenlieferung, wird diese überschrieben.

Parameters

Name	Description
Datentransfernummer	Die Datentransfernummer, die die Transaktion identifiziert

Header

Name	Description
------	-------------



Authorization	JWT Access Token
---------------	------------------

Request Body

Type	Content
APPLICATION/OCTET-STREAM	Die erzeugten Daten entsprechend dem DIP-XML-Schemas. Siehe auch Kapitel 5 Anforderungen an die zu übertragenden Dateien

Responses

Status Code	Description
200	Einreichung erfolgreich
400	Ungültige Datentransfernummer
401	Ungültiger Authorization Header
410	Datentransfer wurde bereits beendet
424	Der Datenübermittler ist dem Vorgang nicht zugeordnet
500	Interner Fehler

6.5.2.4. Anhang übermitteln

PUT /dip/md/{datentransfernummer}/attachment

Einreichung zusätzlicher Anhänge unter Angabe der Datentransfernummer der Massendatenlieferung. Existiert bereits eine Datei für Anhänge zu der Datenlieferung, wird diese überschrieben.

Parameters

Name	Description
datentransfernummer	Die Datentransfernummer, die die Transaktion identifiziert

Header

Name	Description
Authorization	JWT Access Token

Request Body

Type	Content
APPLICATION/OCTET-STREAM	Die Datei mit den Anhängen



Responses

Status Code	Description
200	Einreichung erfolgreich
400	Ungültige Datentransfernummer
401	Ungültiger Authorization Header
424	Der Datenübermittler ist dem Vorgang nicht zugeordnet
500	Interner Fehler

6.5.2.5. Finalisierung einer Massendatenlieferung

PATCH /dip/md/{datentransfernummer}/finish

Erklärung der Fertigstellung der Massendatenlieferung

Parameters

Name	Description
datentransfernummer	Die Datentransfernummer, die die Transaktion identifiziert.

Header

Name	Description
Authorization	JWT Access Token

Responses

Status Code	Description
200	Anfrage erfolgreich
400	Ungültige Datentransfernummer
401	Ungültiger Authorization Header
410	Datentransfer wurde bereits beendet
424	Der Datenübermittler ist dem Vorgang nicht zugeordnet
500	Interner Fehler

6.5.2.6. Abbrechen einer Massendatenlieferung

PATCH /dip/md/{datentransfernummer}/abort

Abbruch der laufenden Übertragung und Verwerfen der bisher übertragenen Dateien.

Parameters

Name	Description
------	-------------



datentransfernummer	Die Datentransfernummer, die die Transaktion identifiziert.
---------------------	---

Header

Name	Description
Authorization	JWT Access Token

Responses

Status Code	Description
200	Anfrage erfolgreich
400	Ungültige Datentransfernummer
401	Ungültige Header Authorization
410	Datentransfer wurde bereits beendet
424	Der Datenübermittler ist dem Vorgang nicht zugeordnet
500	Interner Fehler

6.5.2.7. Abruf des Verarbeitungsprotokolls

GET /dip/md/{datentransfernummer}/protocol

Abruf des Ergebnisprotokolls.

Parameters

Name	Description
datentransfernummer	Die Datentransfernummer, die die Transaktion identifiziert.

Header

Name	Description
Authorization	JWT Access Token

Responses

Status Code	Description
200	APPLICATION/OCTET-STREAM das Verarbeitungsprotokoll
400	Ungültige Datentransfernummer
401	Ungültiger Authorization Header
404	Datei nicht gefunden, Protokoll liegt noch nicht vor



424	Der Datenübermittler ist dem Vorgang nicht zugeordnet
500	Interner Fehler

6.5.2.8. Bestätigung des Abrufs des Verarbeitungsprotokolls

PATCH /dip/md/{datentransfernummer}protocol

Abruf des Protokolls bestätigen.

Parameters

Name	Description
datentransfernummer	Die Datentransfernummer, die die Transaktion identifiziert

Header

Name	Description
Authorization	JWT Access Token

Responses

Status Code	Description
200	Ok
400	Ungültige Datentransfernummer
401	Ungültiges Access Token
404	Protokoll liegt noch nicht vor
424	Der Datenübermittler ist dem Vorgang nicht zugeordnet
500	Interner Fehler

6.5.2.9. Abruf einer Liste verfügbarer Protokolle

GET /dip/md/protocolnumbers

Abruf einer Liste von Datentransfernummern, zu denen ein Protokoll abgerufen

6.5.2.10. Header

Name	Description
Authorization	JWT Access Token



6.5.2.11. Responses

Status Code	Description
200	APPLICATION/OCTET-STREAM <u>Beispiel:</u> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <Datentransfernummern> <Datentransfernummer>ppmca61j9bbu5b1e8npp</Datentransfernummer> </Datentransfernummern>
401	Ungültiges Access Token
500	Interner Fehler

7. Prüfungen und Rückmeldungen der DIP-Schnittstelle

7.1. Eingangsprüfungen einer DIP-Lieferung

Nach Erkennung und Quittierung einer erfolgreichen Massendaten-Lieferung an DIP wird die Massendaten-Datei formalen und inhaltlichen Prüfungen unterzogen. Erst mit dem Erhalt eines positiven Protokolls ist die korrekte Übermittlung einer Massendaten-Lieferung abgeschlossen.

Prüfungen innerhalb der DIP-Schnittstelle

Schritt 1: Prüfung auf Vollständigkeit:

- ➔ Datei(en) mit DIP-XML und Signatur sind beide vorhanden.

Schritt 2: Prüfung der Dateigrößen: Die XML- und ggf. ZIP-Datei sind nicht leer und überschreiten nicht das Limit ihrer maximalen Dateigröße.

Schritt 3: Prüfung des Encodings: Die XML-Datei ist in UTF-8 kodiert.

Schritt 4: Virenprüfung: XML- und Signaturdatei enthalten keinen Virus, optionale ZIP-Datei wird in einem späteren Validierungsschritt überprüft.

Schritt 5: Signaturprüfung:

- ➔ Einlesen und Interpretieren der Signatur ist möglich
- ➔ Übereinstimmungsprüfung der überlieferten Signatur mit der Signatur der XML-Datei



Schritt 6: XML-Schema-Validierung des DipHeaders

Schritt 7: Umgebungsprüfung: Die Angabe unter XPath dip/header/environment entspricht der aktuell angesprochenen Systemumgebung

Schritt 8: Identifizierung des Datenübermittlers kann erfolgreich durchgeführt werden

Schritt 9: Autorisierung des Datenübermittlers führt zu keiner Abweisung, z.B. durch Sperrvermerk des Kontos

Schritt 10: ZIP-Datei-Prüfung, falls eine ZIP-Datei mitgeliefert wurde (Virenprüfung)

Schritt 11: Duplikatsprüfung: die Lieferung wurde nicht wiederholt geliefert

Schritt 12: Referenzprüfung (falls angegeben): die referenceId verweist auf eine bekannte, vorhergehende Lieferung des Kunden/Datenübermittlers

Schritt 13: Routingprüfung: das angegebene Fachverfahren ist bekannt

Prüfung außerhalb der DIP-Schnittstelle

Validierung der Nutzdaten

Die Validierung der fachlichen Nutzdaten (auch Payload genannt) erfolgen **nicht** durch das DIP-Protokoll. Die fachlichen Prüfungen erfolgen durch die Fachverfahren und werden als Ergebnis gesondert im DIP-Protokoll ausgewiesen. Zu den fachlichen Schritten zählen u.a. (keine abschließende Aufzählung):

- Schema-Validierung der Nutzdaten gegen das verwendete fachliche XML-Schema
- Prüfung fachlicher Geschäftsregeln (engl. *business rules*) auf den gelieferten Nutzdaten
- weitere fachliche Validierungen der Nutzdaten

7.2. Ergebnisstatus und Fehlercodierungen

Das DIP-Protokoll liefert als Ergebnis einen Gesamtstatus zurück, der im XML im Element *dipProtocol/processStatus* hinterlegt ist:

- *OK*, falls keine Validierung (weder technisch durch das DIP-Massendatenprotokoll noch fachlich durch das angesprochene Fachverfahren) fehlgeschlagen ist. Das Element *dipResult* ist nicht befüllt, das Element *payloadResult* kann vom Fachverfahren befüllt sein. Dies kann von Fachverfahren zu Fachverfahren



unterschiedlich gehandhabt werden.

- *PARTIALLY_REJECTED*, wenn die Lieferung grundsätzlich angenommen wurde, aber teilweise mit Fehlern/Hinweisen behaftet ist. Details dazu sind im Element *payloadResult* festgehalten und erfolgen durch das Fachverfahren.
- *ERROR*, wenn die Lieferung als Ganzes abgelehnt wurde. In diesem Fall sind entweder Elemente aus *dipResult* oder *payloadResult* befüllt.

Das Element *dipResult* kann nur mit dem Status *ERROR* befüllt sein, wenn bei der (technischen) Validierung durch das DIP-Massendatenprotokoll Fehler aufgetreten sind (z.B. eine fehlerhafte Signatur oder ein Virusfund). Ein Fehler aus der DIP-Validierung besitzt immer einen Fehlercode (XML-Element code), eine Beschreibung (XML-Element *message*) und ggf. weitergehende Informationen (XML-Element *cause*).

Das Element *payloadResult* wird immer nur durch das Fachverfahren befüllt. Dies kann in allen drei Ergebnisstatus der Fall sein.

In der folgenden Auflistung finden sich alle Fehlercodierungen, die durch das DIP-Massendatenprotokoll erkannt werden können. Sie basieren auf den im vorhergehenden Abschnitt benannten technischen Prüfungen der POST 2.0-Massendatenkomponente.

Fehlercode	Bedeutung / Fehlertext	Beschreibung
E0100	Die Dip-XML-Datei ist nicht vorhanden	
E0101	Die Signatur(-datei) ist nicht vorhanden	
E0102	Die Datenlieferung wurde nicht abgeschlossen	Wird automatisch gesetzt, wenn Lieferungen nach mehr als einen Tag nicht beendet wurden.
E0200	Die Dip-XML-Datei ist leer (Größe: 0 Bytes)	
E0201	Die Dip-XML-Datei übersteigt die maximale Dateigröße	
E0203	Die Signaturdatei ist leer (Größe: 0 Bytes)	
E0204	Die Signaturdatei übersteigt die maximale Dateigröße	
E0205	Die ZIP-Datei ist leer (Größe: 0 Bytes)	
E0206	Die Größe der ZIP-Datei übersteigt die maximale Dateigröße	
E0300	Die Dip-XML-Datei war nicht in UTF-8 kodiert	



Kommunikationshandbuch DIP-Standard

Standardisierte Datenübermittlung an das BZSt über die
Massendatenschnittstelle DIP

E0400	In der Dip-XML- oder Signatur-Datei wurde ein Virus gefunden	
E0500	Das Einlesen und Interpretieren der Signatur(-datei) ist fehlgeschlagen	
E0501	Die Signatur(-datei) stimmt nicht mit der Signatur der Dip-XML-Datei überein	<u>Beispiele:</u> <ul style="list-style-type: none">• Digest ungültig• Signatur ungültig• Zertifikat nicht übereinstimmend• Falscher Algorithmus beim Signieren• Zu signierendes Objekt wurde nicht gefunden oder wurde nicht (mit-)signiert• Falscher Algorithmus beim Berechnen des Digests
E0502	Es sind zwei Signaturen vorhanden: enveloping und detached Signatur	
E0600	Die XML-Schema-Validierung ist fehlgeschlagen	
E0601	Die angegebene Schnittstellen-Version ist für den Endpunkt ungültig	
E0602	Die ConsignmentItemPosition ist nicht eindeutig	Alle Consignmentitems einer Lieferung müssen eine eindeutige Nummer haben
E0603	Fehlende XML-Deklaration	XML Dateien müssen mit einer Deklaration beginnen z.B. <code><?xml version="1.0" encoding="UTF-8" ?></code>
E0700	Die Angabe der Umgebung stimmt nicht mit der tatsächlichen Systemumgebung überein	
E0801	Der Übermittler des Dip-XML entspricht nicht dem Erzeuger	Der Fehler E0801 besagt, dass das Ordnungsmerkmal in den XML Daten und das Merkmal, das bei der Registrierung für DIP angegeben wurde nicht übereinstimmen. Im XML muss in <code>header.consignment.customerIdentifier</code> der <ul style="list-style-type: none">• „identityProvider“ (BZST-CERT oder ELSTER)• und die passende Nummer



Kommunikationshandbuch DIP-Standard

Standardisierte Datenübermittlung an das BZSt über die
Massendatenschnittstelle DIP

		(BZST-Nummer oder Steuernummer) als „identifizier“ angegeben sein. Dies müssen die gleichen Werte sein, die bei der Registrierung für DIP über IDP aus dem Zertifikat für die Anmeldung beim Portal übernommen wurden.
E0902	Das Konto des Datenübermittlers ist für das angegebene Fachverfahren gesperrt	
E1000	Fehler bei der Verarbeitung der ZIP-Datei	
E1001	Die ZIP-Datei ist virenbehaftet	
E1100	Duplikatslieferung wurde anhand der Transferticket-ID erkannt	Es wird geprüft, ob die bei der Datenlieferung des Nutzers angegeben Transferticket-ID (die eindeutige ID des Kunden) bereits in einer anderen Datenlieferung verwendet wurde.
E1200	Die Referenz-ID verweist auf eine unbekannte Lieferung	
E1300	Routingprüfung: das angegebene Fachverfahren ist nicht bekannt oder nicht angebunden	Dieser Fehler kann als Body zu einem HTTP-Status 404 nach einem Aufruf des "start" Endpunkts geliefert werden.
E1301	Die Version der Massendatenschnittstelle wird nicht vom Fachverfahren unterstützt	Dieser Fehler kann als Body zu einem HTTP-Status 404 nach einem Aufruf des "start" Endpunkts geliefert werden.
E1302	Das Fachverfahren der Anmeldung stimmt nicht mit dem in der XML-Datei überein	
E9999	Sonstiger Fehler	

Tabelle 12 Fehlercodes DIP

7.3. Abfrage von erstellten Protokollen

Bei der DIP-Schnittstelle können alle vorliegenden Protokollnummern abgefragt werden. Danach kann der Abruf der Protokolle je Nummer erfolgen und die Verarbeitung gestartet werden.

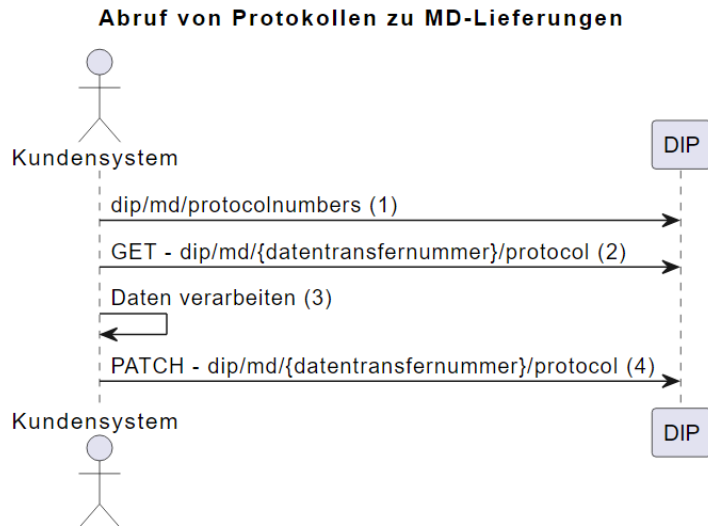


Abbildung 12 Abruf von Protokollen zu MD-Lieferungen

Schritt 1: Abruf aller Transfernummern, für die die Abholung des Protokolls nicht bestätigt wurde

Schritt 2: Abruf des Protokolls zu einer Nummer

Schritt 3: Verarbeitung des Protokolls

Schritt 4: Bestätigung des Protokolls und damit Abschluss des Vorgangs

7.4. Aufbau des Feedback-XML-Schemas Version 1

7.4.1. Inhalte des Elements DipResponse

Das Verarbeitungsprotokoll ist das Ergebnis nach formaler und inhaltlicher Prüfung der Massendaten-Lieferung durch das BZSt. Inhaltlich ist es als XML-Datensatz nach dem dipResponse-XSD aufgebaut.

Im Element *dipProtocol/processStatus* wird der erreichte Ergebniszustand hinterlegt.

Folgende Werte sind möglich:

- OK: Lieferung wurde angenommen und ist fehlerfrei.
- PARTIALLY_REJECTED: die Lieferung wurde angenommen, ist aber teilweise mit Fehlern/Hinweisen behaftet.
- ERROR: die Lieferung wird als Ganzes abgelehnt, Details sind entweder im Element *dipResult* oder im Element *payloadResult*
 - Technische Fehler im Envelop (*dipType*) durch Prüfungen in DIP (Größe, Virus, Signatur, ...): Details sind im Element *dipResult* enthalten



Kommunikationshandbuch DIP-Standard

Standardisierte Datenübermittlung an das BZSt über die
Massendatenschnittstelle DIP

- Fachliche Fehler oder Hinweise im Payload durch Prüfungen im Fachverfahren: Details im Element *payloadResult*

Weitere Details dazu befinden sich im Kapitel [7.2 Ergebnisstatus und Fehlercodierungen](#).

Einige Validierungsschritte können bei Fehlschlägen ihrer Prüfungsbedingung zum vorzeitigen Abbruch weiterer Validierungsschritte führen, da weitere Prüfungen nicht sinnvoll erscheinen (Beispiel: E0200 - Die XML-Datei ist leer). In diesem Fall enthält das Fehlerprotokoll nur alle bis zu diesem Schritt angefallenen Fehler.

Ein *dipProtocol* enthält optional die Angabe des *consignment*-Knotens aus der Lieferung (*dipHeader*) und die Angabe eines *processStatus*-Knotens. Das umschließende *dipResponse*-XML enthält eine von DIP vergebene *responseTransferticketId*, welches die Antwort-XML-Datei eineindeutig identifiziert.

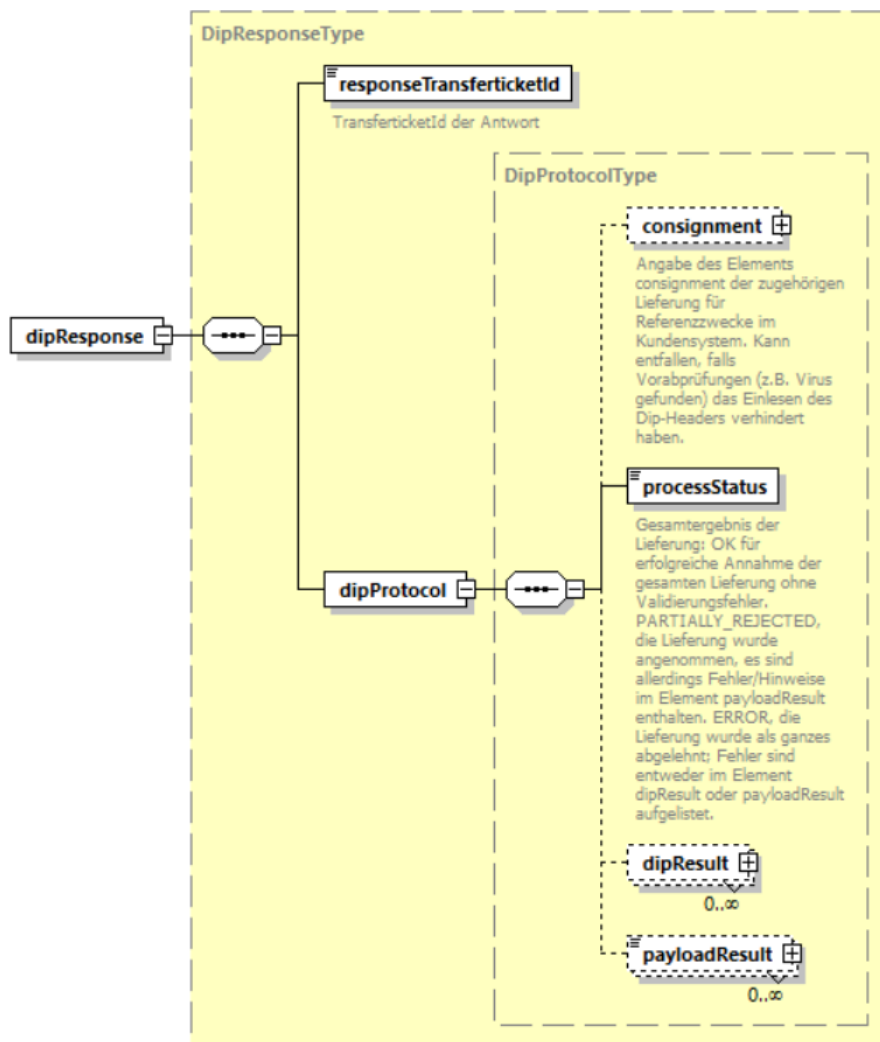


Abbildung 13 XML-Schema *dipResponse* mit *consignment* (Teil 1)



Kommunikationshandbuch DIP-Standard

Standardisierte Datenübermittlung an das BZSt über die
Massendatenschnittstelle DIP

Im Fehlerfall sind Fehler, die durch das DIP-Protokoll entdeckt wurden, in den Elementen *dipResult* protokolliert. Die Prüfschritte und Fehlercodierungen, die das DIP-Protokoll vorsieht, sind in den beiden folgenden Abschnitten beschrieben.

Fehler, die durch fachliche Prüfungen auf den Nutzdaten (Inhalte von *body/consignmentItem/data/* einer Lieferung) erkannt werden, sind in den Elementen *payloadResult* protokolliert. Die *consignmentItemPosition* bezieht sich auf die Position in der Liste der *consignmentItems* bei der Datenlieferung. Das erste *consignmentItem* hat die Position 0.

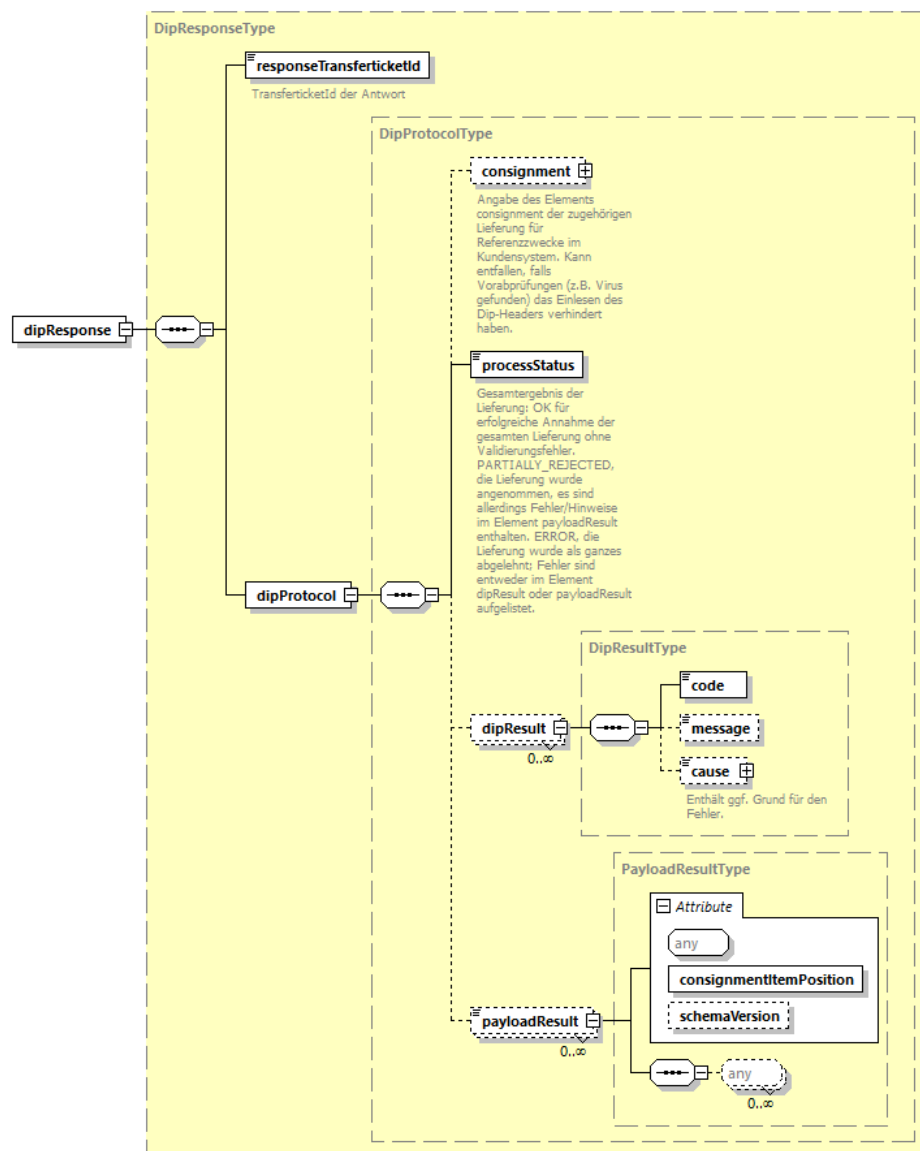


Abbildung 14 XML-Schema *dipResponse* (Teil 2)



7.4.2. Beispiel

Für die folgenden Beispiele wird angenommen, dass die Massendatenlieferung als transferticketId den Wert 58a1-68fd-7777-dcc1-522a besitzt.

Beispiel für eine erfolgreiche Verarbeitung einer Massendatenlieferung:

```
<dipResponse>  
  <responseTransferticketId>0123-4567-89ab-cdef-007a</responseTransferticketId>  
  <dipProtocol>  
    <consignment>  
      <customerIdentifier>  
        <identityProvider>BZST-CERT</identityProvider>  
        <identifier>BZ123456789</identifier>  
      </customerIdentifier>  
      <creationTime>2023-09-01T04:55:22Z</creationTime>  
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>  
    </consignment>  
    <processStatus>OK</processStatus>  
  </dipProtocol>  
</dipResponse>
```

Beispiel für eine Massendatenlieferung mit Validierungsfehlern im DIP-Protokoll:

```
<dipResponse>  
  <responseTransferticketId>0123-4567-89ab-cdef-4711</responseTransferticketId>  
  <dipProtocol>  
    <consignment>  
      <customerIdentifier>  
        <identityProvider>BZST-CERT</identityProvider>  
        <identifier>BZ123456789</identifier>  
      </customerIdentifier>  
      <creationTime>2023-09-01T04:55:22Z</creationTime>  
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>  
    </consignment>  
    <processStatus>ERROR</processStatus>  
    <dipResult>  
      <code>E1100</code>  
      <message>Diese Datei wurde bereits geliefert (Duplikatslieferung erkannt)</message>  
    </dipResult>  
    <dipResult>  
      <code>E1200</code>  
      <message>Die Referenz-ID verweist auf eine unbekannte Lieferung</message>  
    </dipResult>  
  </dipProtocol>  
</dipResponse>
```



Beispiel für eine Massendatenlieferung mit Validierungsfehlern in den fachlichen Nutzdaten (der fachliche Fehlerblock ist nur skizziert und kann sich je nach Leistung/Fachverfahren unterscheiden, hier sind die Fach-XML-Schemata der Fachverfahren zu konsultieren):

```
<dipResponse>
  <responseTransferticketId>0123-4567-89ab-cdef-0815</responseTransferticketId>
  <dipProtocol>
    <consignment>
      <customerIdentifier>
        <identityProvider>BZST-CERT</identityProvider>
        <identifier>BZ123456789</identifier>
      </customerIdentifier>
      <creationTime>2023-09-01T04:55:22Z</creationTime>
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>
    </consignment>
    <processStatus>ERROR</processStatus>
    <payloadResult consignmentItemPosition="0">
      <dsm:DAC7StatusMessage version="1.0" xmlns:dsm="urn:oe.cd:ties:dsm:v1">
        <dsm:MessageSpec>
          <dsm:TransmittingCountry>AF</dsm:TransmittingCountry>
          <dsm:ReceivingCountry>AF</dsm:ReceivingCountry>
          <dsm:MessageType>DPI</dsm:MessageType>
          ...
        </dsm:MessageSpec>
        <dsm:DSMBody>
          ...
          <dsm:ValidationErrors>
            ...
          </dsm:ValidationErrors>
          ...
        </dsm:DSMBody>
      </dsm:DAC7StatusMessage>
    </payloadResult>
  </dipProtocol>
</dipResponse>
```

7.5. **Aufbau des Feedback-XML-Schemas Version 2**

7.5.1. Adresse zum XML-Schema

Das XML-Schema wird Ihnen auf der Homepage des BZSt unter „Massendaten“ zum Download angeboten.



7.5.2. Inhalte des Elements DipResponse

Das Verarbeitungsprotokoll ist das Ergebnis nach formaler und inhaltlicher Prüfung der Massendaten-Lieferung durch das BZSt. Inhaltlich ist es als XML-Datensatz nach dem dipResponse-XSD aufgebaut. Ein *dipResponse* hat ein Attribut – in dem die Version der Schnittstelle zurückgegeben wird, um Version 1.0 und 2.0 unterscheiden zu können.

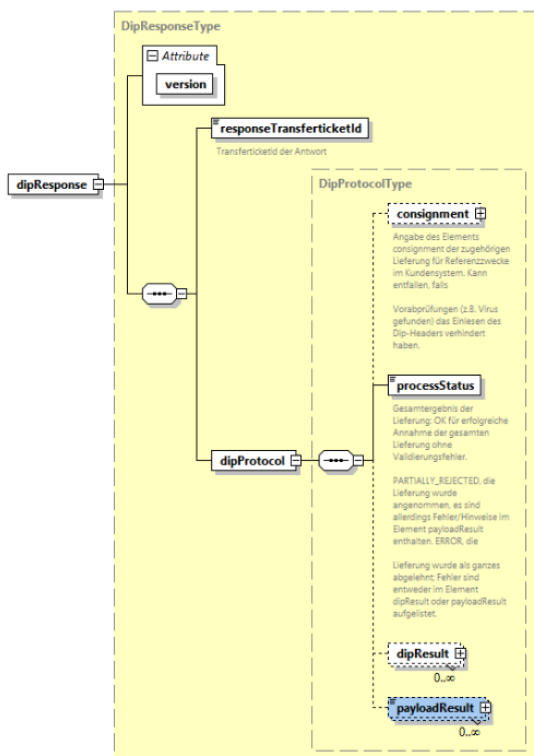


Abbildung 15 XML-Schema dipResponse

Des Weiteren enthält der Response die *responseTransferticketId*, die den Response eindeutig identifiziert und das eigentliche *dipProtocol*, welches im Folgenden erklärt wird.

Im Element *dipProtocol/consignment* werden i.A. die Consignment-informationen der Lieferung zurückgegeben. Folgende Werte sind möglich:

- **OK**: Lieferung wurde angenommen und ist fehlerfrei.
- **PARTIALLY_REJECTED**: die Lieferung wurde angenommen, ist aber teilweise mit Fehlern/Hinweisen behaftet.
- **ERROR**: die Lieferung wird als Ganzes abgelehnt, Details sind entweder im Element *dipResult* oder im Element *payloadResult*
 - Technische Fehler im Envelop (*dipType*) durch Prüfungen in DIP (Größe, Virus, Signatur, ...): Details sind im Element *dipResult* enthalten
 - Fachliche Fehler oder Hinweise im Payload durch Prüfungen im



Fachverfahren: Details im Element *payloadResult*

Weitere Details dazu befinden sich im Kapitel [7.2 Ergebnisstatus und Fehlercodierungen](#).

Der Knoten *dipProtocol/pipResult* wird gesetzt, wenn bei den technischen Validierungsschritten ein Fehler erkannt wurde. Einige Validierungsschritte können bei Fehlschlagen ihrer Prüfungsbedingung zum vorzeitigen Abbruch weiterer Validierungsschritte führen, da weitere Prüfungen nicht sinnvoll erscheinen (Beispiel: E0200 - Die XML-Datei ist leer). In diesem Fall enthält das Fehlerprotokoll nur alle bis zu diesem Schritt angefallenen Fehler.

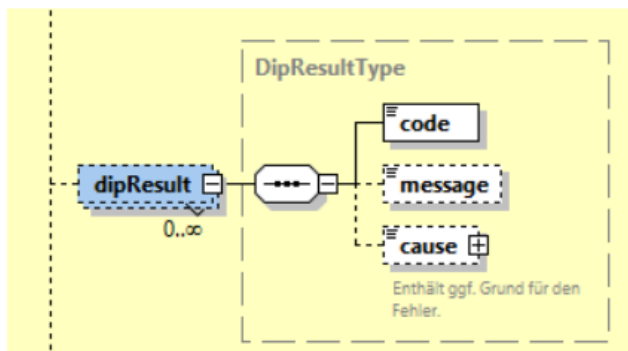


Abbildung 16 XML-Schema dipResponse mit consignment (Teil 1)

Das Element *dipProtocol/payloadResult* wird gesetzt, wenn keine technischen Fehler aufgetreten sind. Es enthält die Ergebnisse der fachlichen Prüfungen der Nutzdaten (Inhalte von *body/consignmentItem/data/* der Lieferung). Die *consignmentItemPosition* bezieht sich auf die *consignmentItemPosition* in der Liste der *consignmentItems* bei der Datenlieferung und verknüpft die Lieferung und das Ergebnis. Jedes Result bezieht sich auf eine Position und enthält das jeweilige Ergebnis-XML.

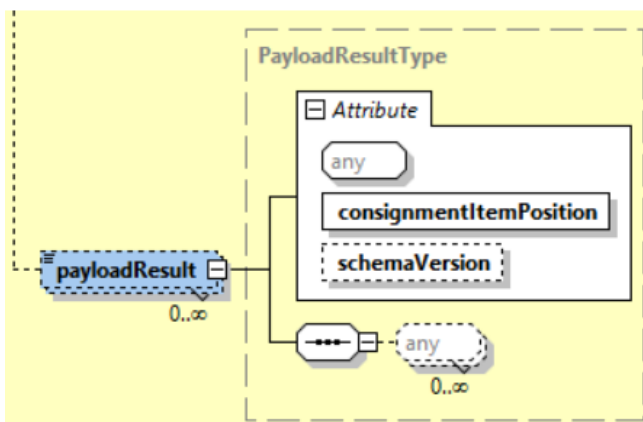


Abbildung 17 XML.Schema dipResponse (Teil 2)



7.5.3. Beispiel

Für die folgenden Beispiele wird angenommen, dass die Massendatenlieferung als *transferticketId* den Wert *58a1-68fd-7777-dcc1-522a* besitzt.

Beispiel für eine erfolgreiche Verarbeitung einer Massendatenlieferung:

```
<dipResponse>  
  <responseTransferticketId>0123-4567-89ab-cdef-007a</responseTransferticketId>  
  <dipProtocol>  
    <consignment>  
      <customerIdentifier>  
        <identityProvider>BZST-CERT</identityProvider>  
        <identifier>BZ123456789</identifier>  
      </customerIdentifier>  
      <creationTime>2023-09-01T04:55:22Z</creationTime>  
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>  
    </consignment>  
    <processStatus>OK</processStatus>  
  </dipProtocol>  
</dipResponse>
```

■ Ein Payloadresult muss nicht existieren. Einzelne Fachverfahren können auch nur den *processStatus* zurückliefern.

Beispiel für eine Massendatenlieferung mit Validierungsfehlern im DIP-Protokoll:

```
<dipResponse>  
  <responseTransferticketId>0123-4567-89ab-cdef-4711</responseTransferticketId>  
  <dipProtocol>  
    <consignment>  
      <customerIdentifier>  
        <identityProvider>BZST-CERT</identityProvider>  
        <identifier>BZ123456789</identifier>  
      </customerIdentifier>  
      <creationTime>2023-09-01T04:55:22Z</creationTime>  
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>  
    </consignment>  
    <processStatus>ERROR</processStatus>  
    <dipResult>  
      <code>E1100</code>  
      <message>Diese Datei wurde bereits geliefert (Duplikatslieferung erkannt)</message>  
    </dipResult>  
    <dipResult>  
      <code>E1200</code>
```



```
<message>Die Referenz-ID verweist auf eine unbekannte Lieferung</message>
</dipResult>
</dipProtocol>
</dipResponse>
```

Beispiel für eine Massendatenlieferung mit Validierungsfehlern in den fachlichen Nutzdaten (der fachliche Fehlerblock ist nur skizziert und kann sich je nach Leistung/Fachverfahren unterscheiden, hier sind die Fach-XML-Schemata der Fachverfahren zu konsultieren):

```
<dipResponse>
  <responseTransferticketId>0123-4567-89ab-cdef-0815</responseTransferticketId>
  <dipProtocol>
    <consignment>
      <customerIdentifier>
        <identityProvider>BZST-CERT</identityProvider>
        <identifier>BZ123456789</identifier>
      </customerIdentifier>
      <creationTime>2023-09-01T04:55:22Z</creationTime>
      <transferticketId>58a1-68fd-7777-dcc1-522a</transferticketId>
    </consignment>
    <processStatus>ERROR</processStatus>
    <payloadResult consignmentItemPosition="0">
      <dsm:DAC7StatusMessage version="1.0" xmlns:dsm="urn:oe.cd:ties:dsm:v1">
        <dsm:MessageSpec>
          <dsm:TransmittingCountry>AF</dsm:TransmittingCountry>
          <dsm:ReceivingCountry>AF</dsm:ReceivingCountry>
          <dsm:MessageType>DPI</dsm:MessageType>
          ...
        </dsm:MessageSpec>
        <dsm:DSMBody>
          ...
          <dsm:ValidationErrors>
            ...
          </dsm:ValidationErrors>
          ...
        </dsm:DSMBody>
      </dsm:DAC7StatusMessage>
    </payloadResult>
  </dipProtocol>
</dipResponse>
```



8. Zusätzliche Informationen

8.1. Sperren des Zugangs zur Massendatenschnittstelle

Der Zugang zur DIP-Schnittstelle kann technisch für den Nutzer global gesperrt werden. Kunden können eine Sperre - z.B., wenn das Zertifikat kompromittiert wurde - über das BZSt anfordern. Das Thema Deaktivierung des Zugangs ist in Kapitel [3 Freischaltung, Aktualisierung und Deaktivierung der DIP Schnittstelle](#) beschrieben.

8.2. Vorgehen bei Ablauf des in der MDS hinterlegten Zertifikats

Sollte das vom Kunden hinterlegte Zertifikat seine Gültigkeit verlieren, muss für den Austausch ein Änderungsantrag gestellt werden (siehe Kapitel [3 Freischaltung, Aktualisierung und Deaktivierung der DIP Schnittstelle](#)). In dem Änderungsantrag müssen alle Felder mit den aktuell gültigen Werten angegeben werden (z.B. auch die Kontakt-Emailadresse), da bei einem Änderungsantrag alle Daten ersetzt werden.

8.3. Änderung des Ansprechpartners

Eine Änderung des Ansprechpartners bzw. der Kontaktmailadresse erfolgt über den Änderungsantrag. Wie bei der Erneuerung des Zertifikats im vorhergehenden Kapitel müssen auch hier **alle** Felder des Antrags mit den aktuell gültigen Werten belegt werden (insb. auch das Zertifikat), da bei einem Änderungsantrag alle Daten ersetzt werden.



9. Abkürzungsverzeichnis/Glossar

Anfangsbuchstabe	Abkürzung	Erläuterung
C	CA	<i>certificate authority</i> oder <i>certification authority</i> , Bezeichnung einer Zertifizierungsstelle, welche digitale Zertifikate herausgibt und beglaubigt.
	cURL	<i>client for URLs</i> oder <i>curl URL request library</i> , Bezeichnung eines Kommandozeilen-Programms und einer Programmbibliothek, welche den Versand von Dateien in Rechnernetzen ermöglicht.
D	DIP	Digitaler Posteingang, Komponente des BZSt online.portal (BOP-neu), welche die Schnittstelle zur Überlieferung von Massendaten bereitstellt.
J	JSON	<i>JavaScript Object Notation</i> , Bezeichnung eines textbasierten Datenformats zum Austausch von Daten zwischen Anwendungen
	JWT	<i>JSON Web Token</i> , Bezeichnung eines auf JSON basierenden Access-Tokens
K	KHB	Kommunikationshandbuch
	KTST	Kudentest-Stellung, Umgebung für technische Tests des Kunden
M	MDS	Massendatenschnittstelle
P	PROD	produktive Umgebung, Bezeichnung der Umgebung, auf welcher das BZSt online.portal (BOP-neu) zur produktiven Verwendung durch die Nutzenden bereitgestellt wird.
R	REST	<i>representational state transfer</i> , Modell zur Beschreibung der Softwarearchitektur von verteilten Systemen, welches beispielsweise zur Beschreibung einheitlicher Schnittstellen zur Interaktion mit einer Web-Anwendung eingesetzt wird.
U	URL	<i>uniform resource locator</i> , Bezeichnungsstandard für Netzwerkressourcen

Tabelle 13 Abkürzungsverzeichnis/Glossar

Impressum

Herausgeber:

Bundeszentralamt für Steuern
An der Kuppe 1
53225 Bonn
Telefon: +49 228 406-0
Internet: www.bzst.bund.de

Stand:

Version 2.1, 01/2025

Bildnachweis:

Titelseite: Hardy Welsch (<http://www.hardy-welsch.de>)

Text:

BZSt