Bundesministerium
des Innern
und für Heimat

bundID

# Schnittstellenbeschreibung

## Zentrales Bürgerpostfach

23. Mai 2024

# Überblick über das Dokument

| | |
|---|---|
| Name des Dokuments | ZBP_Schnittstellenbeschreibung<br>Zentrales Bürgerpostfach |
| Dokument-Version | 1.0 |
| Release-Version | 3.0.0.0 |
| API-Version | V6 |
| Zweck des Dokuments | Beschreibung der REST-Schnittstelle für das Zentrale Bürgerpostfach (ZBP) |
| Herausgebende Stelle | Bundesministerium des Innern und für Heimat<br>DVI5 |
| Stand | 23.05.2024 |

# Versionshistorie

| Version | Beschreibung | Datum |
|---|---|---|
| 1.5 | N/A | 12.12.2023 |
| 1.6 | N/A | 31.01.2024 |
| 1.7 | N/A | 27.02.2024 |

**Bitte beachten Sie, dass ab Mai 2024 eine neue Form der Dokumenten-Versionierung angewendet wird. Fortan gilt die Release-Version der Software als führend. Die Dokument-Version bezieht sich auf die für das Release geltende Schnittstellenbeschreibung und beginnt somit bei einem neuen Major-Release mit Version 1.0. Sollte es innerhalb eines Major-Releases zu Anpassungen in der Schnittstellenbeschreibung kommen, folgt Dokument-Version 2.0.**

| Release-Version | Dokument-Version | Beschreibung | Datum |
|---|---|---|---|
| 3.0.0.0 | 1.0 | Überarbeitung Kapitel 6, 6.1, 6.2.6, 7.1, 7.2 (Metadaten), 8.1, 8.2, 9, 9.3, 9.4, 13, 14, Erweiterung Kapitel 6.2.7, 6.2.8, Entfernung Kapitel 7.2 (Zusätzliche Metadaten) und 8.3 | 23.05.2024 |

# Inhaltsverzeichnis

# 1. Präambel

Die Schnittstellenbeschreibung des Zentralen Bürgerpostfachs (ZBP) beschreibt zwei grundlegende Aspekte:

1. **Authentifizierung**

   → Die Authentifizierung und Autorisierung erfolgt im ZBP mittels JSON Web Token (JWT), das mit einem PKI-Zertifikat signiert wird.

2. **Funktionsumfang der ZBP-Schnittstelle**

   → Es werden nur die Funktionen aus der aktuellsten API-Version beschrieben. Für die Beschreibung älterer API-Versionen dient die jeweilige Schnittstellenbeschreibung der älteren Versionen.

   → Die aktuelle API-Version ist: **API v6**

**Die technische Spezifikation der Schnittstelle wird als eine OpenAPI JSON-Datei zur Verfügung gestellt und ist nicht Teil dieses Dokuments.**

# 2. Erläuterung Begriffe

| Begriff der Dokumentation | Synonyme | Kürzel | Erläuterung |
|---|---|---|---|
| Autor | Verfasser, Ersteller der Nachricht oder Statusmeldung | | Ist die Drittanwendung, die tatsächlich die Nachricht oder Statusmeldung erstellt hat. |
| Bürger | der Benutzer, der Anwender | | |
| Brückenkopf | Übermittler, Message-Hub | BK | Ein Brückenkopf bündelt die Kommunikationsstrecken einer oder mehrerer Drittanwendungen in das ZBP. |
| Drittanwendung | Fachverfahren, Fachdienst, Dienstleistungen, Online-Leistungen, Onlinedienst | DA | Bei Drittanwendung wird nicht unterschieden, welchen Dienst/welche Aufgabe die andere Anwendung hat. Es ist nur wichtig, dass diese Anwendung in irgendeiner Form in Kontakt mit dem ZBP steht. |
| Fachanwendung | Fachanwendung, Behörde | FA | Die Fachanwendung ist das IT-Verfahren (die IT-Anwendung), das in der Behörde läuft und von Sachbearbeitern der Behörde verwendet wird. Hier wird die Bearbeitung des Antrags durchgeführt. |
| Fachdienst | Dienstleistungen, Online-Leistungen, Onlinedienst | FD | Der Fachdienst ist die Webseite, auf der der Bürger seinen Antrag stellt. |
| Fachverfahren | Fachverfahren, Behörde | FV | Das Fachverfahren ist das IT-Verfahren (die IT-Anwendung), das in der Behörde läuft und von Sachbearbeitern der Behörde verwendet wird. Hier wird die Bearbeitung des Antrags durchgeführt. |

| | | | |
|---|---|---|---|
| Nachrichtenformat | | NF | Beschreibt das Transportformat der Nachrichten. Bspw. ZBP, XÖV, FIM |
| Nachrichtenprotokoll | | NP | Beschreibt das Transportprotokoll der Nachrichten. Bspw. REST, OSCI, FIT-Connect |
| Nutzerkonto | BundID, Mandant | NK | |
| Onlinedienst | Fachdienst, Dienstleistungen, Online-Leistungen, Onlinedienst | OD | Der Onlinedienst ist die Webseite, auf welcher der Bürger seinen Antrag stellt. |
| Postfach-Handle | Postkorb-Handle, mailboxId, mailboxUuid | | Ist der eineindeutige Identifikator für ein Postfach im ZBP. |
| Sender | Einsteller der Nachricht oder Antragstatusmeldung in das ZBP | | Ist entweder die Drittanwendung, die eine Nachricht oder Antragstatusmeldung eigenständig in das ZBP einstellt, oder es ist ein Brückenkopf, der im Auftrag einer Drittanwendung die Einstellung einer Nachricht oder Antragstatusmeldung in das ZBP übernimmt. |
| Vertrauensniveau | STORK-QAA-Level | VN | |
| Zentrales Bürgerpostfach | | ZBP | |

# 3. Authentifizierung

Die Authentifizierung und Autorisierung erfolgt im ZBP mittels TLS-Client-Zertifikat und JSON Web Token (JWT). Für jede Anfrage (Request) an das ZBP muss dabei

- das PKI-Zertifikat als TLS-Client-Zertifikat beim Verbindungsaufbau genutzt werden und
- das JWT in Form eines Bearer-Token-Strings als Wert für den Request-Header “Authorization” der HTTPS-Anfrage übermittelt werden.

## 3.1 JSON Web Token (JWT)

Für den Signierungsalgorithmus des JSON Web Token (JWT) muss der RSA512 Algorithmus sowie der private Schlüssel (Private Key) des PKI-Zertifikat eingesetzt werden. Zudem muss der Common Name (CN) aus dem PKI-Zertifikat als “Signer” im JWT angegeben sein.
Die Token-Lebenszeit (= Ablaufdatum - Erstellungsdatum) darf maximal 30 Minuten betragen.

Folgende Fehlerszenarien sind im Bezug auf die Token-Gültigkeitsdauer möglich:

1. Token abgelaufen: Ein Zugriff mit abgelaufenem Token wird abgelehnt und erhält die Fehler-Antwort "*ZBP_401_003: Token expired.*".
2. Token-Gültigkeitsdauer zu lang: Ein nicht abgelaufener Token, der mit einer zu langen Token-Gültigkeitsdauer erstellt worden ist, wird abgelehnt und erhält die Fehler-Antwort "*ZBP_401_004: Token lifetime too long.*".
3. Token noch nicht gültig: Ein Token, dessen Erstellungsdatum in der Zukunft liegt, wird abgelehnt und erhält die Fehler-Antwort "*ZBP_401_005: Token is not valid yet*."

**Übersicht der Angaben für die Erstellung des JSON Web Token:**

| | |
|---|---|
| createdDate | Erstellungsdatum |
| expiredDate | Ablaufdatum |
| Claim »signer« | Common Name (CN) des ZBP PKI-Client-Zertifikats |
| Claim »roles« | Der Inhalt »["THIRD_PARTY"]« |

Das erstellte JWT muss für jede Anfrage (Request) an das ZBP in Form eines Bearer-Token-Strings als Wert für den Request-Header “Authorization” gesetzt werden:

- Beispiel: --header ‘Authorization: Bearer eyJhbGciOiJSUzUxMiIsIn…’

Beispiele für die Erstellung können unter Beispiele zur Erstellung des JSON Web Token eingesehen werden.

## 3.2 PKI-Zertifikat

Das PKI-Zertifikat, d.h. der private Schlüssel (Private Key) und das Zertifikat mit dem öffentlichen Schlüssel (Public Key), werden ab Juni/Juli 2024 zentral durch das Self Service Portal (SSP) ausgegeben. Um Zugang zum SSP zu erhalten, wird ein ELSTER Organisationszertifikat benötigt. Über das SSP können neben produktiven Live-PKI-Zertifikaten auch Test-PKI-Zertifikate (sog. Referenz-PK) bezogen werden. Jedes PKI-Zertifikat und jeder private Schlüssel (Private Key) ist über das SSP mit dem ZBP Aussteller-CA signiert worden und erfüllt gewisse Vorgaben. Die Verwendung eines nicht vom SSP erstellten PKI-Zertifikats und privaten Schlüssels ist nicht möglich.

Das PKI-Zertifikat wird ebenfalls als Client-Zertifikat für den Aufbau der SSL-/TLS-Verbindung verwendet und muss in jeder Anfrage (Request) an das ZBP als TLS-Client-Zertifikat mitgeschickt werden.

Sollten Sie das ZBP bereits vor Verfügbarkeit des SSP (voraussichtlich ab Juni/Juli 2024) testen wollen, dann kann das PKI-Zertifikat manuell für Sie erstellt werden. In diesem Fall wenden Sie sich mit Ihrem Anliegen bitte an: BundID@bmi.bund.de

# 4. Funktionsumfang der Empfangsschnittstelle

## 4.1 Funktionsliste (API v6)

| Funktion | HTTP-Methode + Endpunkt | Authentifizierung | Hinweise |
|---|---|---|---|
| Nachrichten einstellen | *PUT /v6/mailbox/ messages* | PKI-Zertifikat und Security-Token (JWT) | Fachdienste und Fachverfahren können Nachrichten auf einer dedizierten Schnittstelle in ein Postfach eines Bürgers einstellen. Das Nutzerkonto wird über den Eingang einer neuen Nachricht informiert und kann im Folgenden den Bürger per E-Mail über den Eingang benachrichtigen. |

| | | | |
|---|---|---|---|
| Antragsstatusmeldung einstellen | *POST /v6/mailbox/ applications/ states* | PKI-Zertifikat und Security-Token (JWT) | Fachdienste und Fachverfahren können Statusmeldungen zu einem Antrag mit Angabe der Antrags-ID einstellen.<br>Es können mehre Statusmeldungen desselben Status eingestellt werden. Dies dient dazu, um mittels der anderen Felder der Statusmeldung zusätzliche Informationen an den Bürger senden zu können (bspw. das Feld "Weitere Information").<br>Das Nutzerkonto wird über den Eingang einer neuen Statusmeldung informiert und kann im Folgenden den Bürger per E-Mail über den Eingang benachrichtigen. |

## 4.2 Funktionsbeschreibung

### 4.2.1 Nachrichten einstellen

Um eine Nachricht in das ZBP einzustellen, muss der Sender ein gültiges PKI-Zertifikat sowie den entsprechenden privaten Schlüssel besitzen. Beide dienen sowohl der Authentifizierung des Senders als auch der Signierung des Nachrichteninhalts.

Die Identifikation des spezifischen Postfachs erfolgt über die eindeutige Kennung „mailboxUuid“ (auch als Postkorb-Handle bezeichnet). Diese Kennung wird in den Nachrichteninhalt integriert, sodass sie Teil der digitalen Signatur der Nachricht wird.

Die Nachrichtenschnittstelle selbst arbeitet synchron. Das heißt, dass der Empfang und die Prüfung der Signatur direkt durchgeführt werden. Antwortet die Schnittstelle mit HTTP-Statuscode 200 (OK), gilt

die Nachricht als empfangen und geprüft. Der Sender kann sich auf den erfolgreichen Empfang verlassen und erfüllt damit die Bekanntgabefiktion.

Für neue Nachrichten werden jeweils Eingangsbenachrichtigungs-E-Mails über das Nutzerkonto an den betreffenden Bürger ausgelöst. Das ZBP verschickt selbst **keine** Eingangsbenachrichtigungen an Bürger. Das ist die Aufgabe des Nutzerkontos.

Die Nachrichtenschnittstelle kann für dieselbe Nachricht mehrfach aufgerufen werden. Durch Prüfung der Signatur stellt das ZBP fest, ob exakt dieselbe Nachricht bereits im System des ZBP persistiert wurde. In diesem Fall antwortet die Schnittstelle ebenfalls mit HTTP-Statuscode 200 (OK). Allerdings wird die Nachricht nur einmal im ZBP angelegt, wodurch keine Nachrichten-Duplikate vorkommen können.

Es gelten aktuell folgende Limitierungen:

Die Größe der Nachrichten (das Feld content im Nachrichteninhalt) ist auf die Maximalgröße von 1 MB limitiert.

Die maximal zulässige Gesamtgröße der Anhänge liegt bei 25 MB. Ein einzelner Anhang darf ebenfalls bis zu 25 MB groß sein. Die maximale Anzahl an Anhängen ist auf 200 beschränkt. Zulässige Dateiformate für Anhänge sind: pdf, gif, jpg, jpeg, png, svg, tiff, tif, txt, ics, ical, ifb, bmp, rtf, csv.

#### 4.2.1.1 Nachricht einstellen über Brückenkopf

Das ZBP ist ein REST-Service mit JSON als Nachrichtenformat. Alternativ können Fachdienste und Fachverfahren Nachrichten an Bürger auch mittels anderer Nachrichtenprotokolle (bspw. SOAP, OSCI) und Nachrichtenformaten senden. Dafür müssen die Nachrichten in das ZBP über einen sogenannten Brückenkopf als Übersetzer eingestellt werden. Der Brückenkopf empfängt dazu die Nachricht im Format des Fachdienstes oder -verfahrens, übersetzt sie in das vom ZBP erwartete Format und sendet sie dann weiter an das ZBP. Da das ZBP Anfragen (Requests) synchron beantwortet, erhält der Brückenkopf direkt die Antwort (Response) vom ZBP und kann diese wiederum in das Format des entsprechenden Fachdienstes oder -verfahrens übersetzen und an sie weitergeben.

In dieser Nachrichten-Transportkette übernimmt der Brückenkopf die Rolle des Nachrichtensenders und der Fachdienst oder das Fachverfahren weiterhin die Rolle des Nachrichtenautors (Nachrichtenerstellers). Beide Rollen müssen jeweils über ein gültiges PKI-Zertifikat und den entsprechenden privaten Schlüssel verfügen, um sich gegenüber dem ZBP zu authentifizieren und um den Nachrichteninhalt zu signieren.

Für eine Anfrage an das ZBP muss der Brückenkopf als Sender, wie im Abschnitt JSON Web Token (JWT) erläutert, den Bearer-Token-String als Wert für den Request-Header “Authorization” setzen, sein PKI-Zertifikat im PEM-Format für den SSL- / TLS-Verbindungsaufbau angeben und mit seinem privaten Schlüssel aus dem PKI-Zertifikat die Signatur der Nachricht vornehmen und über das Feld “sha512sum” im Nachrichtenobjekt übergeben.
Der Fachdienst/das Fachverfahren muss als Autor der Nachricht gegenüber dem ZBP beweisen, dass er ein valider Autor ist. Dazu muss dieser seinen JSON Web Token (JWT) in Form eines Token-String und sein PKI-Zertifikat mit enthaltenem öffentlichen Schlüssel (Public Key) dem Brückenkopf in seiner Anfrage mitteilen. Der Brückenkopf wiederum muss diese Werte als Sender der Nachricht über die Felder “authorToken” und “authorCertificate” im Nachrichtenobjekt an das ZBP weitergeben.

Details zur Nutzung eines Brückenkopfes müssen individuell mit dem jeweiligen Brückenkopf geklärt werden. Jeder Brückenkopf kann eigene Anforderungen an die Nachrichten definieren.

#### 4.2.1.2 Das Nachrichtenobjekt

Eine Nachricht ist ein Objekt im JSON-Format, das aus den nachfolgend beschriebenen Feldern besteht. Die zwingend erforderlichen Pflichtfelder sind mit **(*)** gekennzeichnet:

- **content (*)**
    - Beschreibung: Der eigentliche Inhaltsteil der Nachricht. Der Wert dieses Feldes muss als ein JSON-String formatiert sein. Das bedeutet, dass alle Daten, die in diesem content-Feld enthalten sind (optionale und Pflichtfelder), als ein String übermittelt werden, der die Struktur eines JSON-Objektes hat. Hierbei ist es wichtig, dass Anführungszeichen und eventuell andere spezielle Zeichen richtig escaped (maskiert) werden, um die Syntax des umgebenden JSON-Formats nicht zu stören. In JSON-Strings müssen Anführungszeichen, die als Teil des Inhalts und nicht als Begrenzer des Strings dienen, mit einem Backslash (\) versehen werden. Zum Beispiel wird "title":"Nachrichtenbetreff" zu \"title\":\"Nachrichtenbetreff\". Der gesamte Inhalt wird als ein String behandelt, was bedeutet, dass der Anfang und das Ende des JSON-Objektes ebenfalls in Anführungszeichen gesetzt und escaped werden müssen.
    - Format: String

- Beispiel:
  "{\"mailboxUuid\":\"0f0407c5-7f7d-4ada-8dfe-43760d90586d\", \"stork_qaa_level\":1, \"sender\":\"Bundeswahlscheinverfahren\", \"title\":\"Nachrichtenbetreff\", \"content\":\"Nachrichtentext\", \"service\":\"Wahlschein-Service\", \"retrievalConfirmationAddress\":\"info@example.com\", \"replyAddress\":\"info@bund.de\", \"attachments\":[{\"filename\":\"Wahlschein.pdf\", \"sha512sum\":\"f3b3ab3e6351e25b5c1882bea8d37efaddc0ea72bf153bb067688f775a26810d32b54f014bf1cebc7fe93042d85b18b5b453e322d154bc55d5cc2754b0dfb4b2\", \"contentLength\":1048576}], \"reference\":\"80364006\", \"senderUrl\":\"https://www.bund.de \", \"applicationId\":\"e8bc5ebc-56cf-4ff0-a861-f989aa1ba4a9\"}"

- **sha512sum (*)**
  - Beschreibung: Ist die Signatur der Nachricht. Der Wert muss mittels dem privaten Schlüssel der PKI und mit dem RSA512 Algorithmus über den JSON-String des "content"-Felds gebildet werden.
  - Format: String
  - Beispiel:
    "mrK2ar3kWk6nvrWc8CeUzdAOyHpQgy5rk8Xi/hTWH4ZiRtmQcmfB4pqKNIiQJcqVqfoXqDix1TaD2RSWYPb2De2HxeY0myD+5Go44V+kTxA9Wl1NTYLQZgOHWWJfkMDDHhRREv5537pUhL/l2RPobQBlEakwiQpzob4iXHzV1tWFOdNw6KnnMX/tou9sGFOaMjgD1HNmEmxIzFDm+gz8trBq/pfOHxMYludkGpsbeUJzFATqMP7lhGVZ0dwj6+2N/YDQe44YS+9hl1LQ6qGgWP5HOZw2XN8Z7SyMqD/Hi1D3hij9rB0GfTxeEBBWhLBIbLYrvad6SzVMeDI+U9LIAZVNiOYJDAlJNspKldHZt5Kda9Wm46WRteNl1Di7kpTbaHA/9ONFX/qGrr4WdaEPghpGhPm9SSNbmfx5FyyDXWYSw6DiDg80ZouA39jp2rBluXB4kjmQ2psjm4BCQ6LKd9yipEj3x72HYmKsC8ueqpyWuceu9DBitARFsw/wvl3HJb9NfV35s1vM0JkJLHHsH1CUifbrJR14UZonAYRRZE-AZAumstSw/bCcQ20MWCqcZ9njBDhmqJdxrYMtGIeyn+ElMaWNXS2q6gnCnbvaskoK/GMJjPpGWUwp89KD13DjEdXoej0PLfUKS7NjW9wceZxT4o1wBjYRMY4yxhNKUCS8="

- **authorToken**
    - Beschreibung: Ist das JSON Web Token (JWT) des Autors bei Verwendung eines Brückenkopfs in Form eines Token-String. Dieser Wert muss angegeben werden, wenn die Nachricht über einen Brückenkopf an das ZBP gesendet wird und somit der Nachrichtensender und Nachrichtensignierer nur der Übermittler (Brückenkopf) und nicht der eigentliche Nachrichtenautor (Drittanwendung) ist.
    - Hinweis: Dieses Feld steht in Abhängigkeit zum Feld “authorCertificate”. Das bedeutet, dass immer auch ein valider Wert für “authorCertificate” angegeben werden muss, sobald ein Wert für “authorToken” übermittelt wird. Dies ist notwendig, damit überprüft und sichergestellt werden kann, dass der Nachrichtenautor auch über den privaten Schlüssel (Private Key) des PKI-Zertifikats verfügt.
    - Format: String
    - Beispiel: “eyJhbGciOiJSUzUxMilsIn….”
- **authorCertificate**
    - Beschreibung: Ist das PKI-Zertifikat des Autors als PEM-Format-String. Dieser Wert muss angegeben werden, wenn die Nachricht über einen Brückenkopf an das ZBP gesendet wird und somit der Nachrichtensender und -signierer nur der Übermittler (Brückenkopf) und nicht der eigentliche Nachrichtenautor (Drittanwendung) ist.
    - Hinweis: Dieses Feld steht in Abhängigkeit zum Feld “authorToken”. Das bedeutet, dass immer auch ein valider Wert für “authorToken” angegeben werden muss, sobald ein Wert für “authorCertificate” übermittelt wird. Dies ist notwendig, damit überprüft und sichergestellt werden kann, dass der Nachrichtenautor auch über den privaten Schlüssel (Private Key) des PKI-Zertifikats verfügt.
    - Format: String
    - Beispiel: "-----BEGIN CERTIFICATE-----MII-FIjCCAwoCFBRs6KrFt2pgOK9G750NZZ6aQ6v+MA[…]B0rdG84mEM9oQ==-----END CERTIFICATE-----"

#### 4.2.1.3 Der Nachrichten-Inhalt

Eine Nachricht unterstützt im Inhaltsteil folgende fachliche Parameter, wobei zwingend erforderliche Pflichtfelder mit **(*)** gekennzeichnet sind:

- **mailboxUuid (*)**
    - Beschreibung: Ist die Postfach-ID (auch Postkorb-Handle genannt) des Empfängers (Bürgers).
    - Format: UUID String
    - Beispiel: “0f0407c5-7f7d-4ada-8dfe-43760d90586d”
- **sender (*)**
    - Beschreibung: Der ursprüngliche Absender der Nachricht (bspw. Name des Fachverfahrens oder der Behörde)
    - Format: String
    - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 255 Zeichen
    - Beispiel: OK.VERKEHR
- **title (*)**
    - Beschreibung: Text der Betreffzeile einer Nachricht
    - Format: String
    - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 1024 Zeichen
- **content (*)**
    - Beschreibung: Ist der Nachrichtentext
    - Format: String
    - Limitierung: darf nicht “Null” sein.
- **service (*)**
    - Beschreibung: Name des Dienstes (bspw. Bezeichnung des Fachverfahrens)
    - Format: String
    - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 255 Zeichen
    - Beispiel: “Kfz-Zulassungsstelle”

- **senderUrl**
    - Beschreibung: Ist die URL des Absenders
    - Format: String
    - Limitierung: Maximallänge = 255 Zeichen
    - Beispiel: "https://www.example.de"
- **reference**
    - Beschreibung: Ist die Referenz / das Aktenzeichen zu einem Vorgang
    - Format: String
    - Limitierung: Maximallänge = 255 Zeichen
    - Beispiel: “AZ-123456”
- **retrievalConfirmationAddress**
    - Beschreibung: Ist die Adresse der Drittanwendung als Empfänger für die Lesebestätigung
    - Format: String
    - Limitierung: Maximallänge = 320 Zeichen
    - Beispiel: “confirm@example.de”
- **replyAddress**
    - Beschreibung: ist die Antwortadresse der Drittanwendung
    - Format: String
    - Limitierung: Maximallänge = 320 Zeichen
    - Beispiel: “reply@example.de”
- **applicationId**
    - Beschreibung: Referenz / Verknüpfung zu einem Antrag (AntragsID)
    - Format: UUID String
    - Beispiel: “e8bc5ebc-56cf-4ff0-a861-f989aa1ba4a9”

- **attachments**
    - Beschreibung: Liste von Objekten mit Metadaten der Dateianhänge
    - Format: Array mit Anhangsobjekten
    - Limitierung: Jedes Anhangsobjekt muss folgende Metadaten enthalten:
        - **filename**
            - Beschreibung: Name der Anhangsdatei
            - Format: String
            - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 4000 Zeichen
            - Beispiel: “Wahlschein.pdf”
        - **sha512sum**
            - Beschreibung: SHA-512 Checksumme der Anhangsdatei
            - Format: String
            - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 128 Zeichen
            - Beispiel: "f3b3ab3e6351e25b5c1882bea8d37efaddc0ea72bf153bb067688f775a26810d32b54f014bf1cebc7fe93042d85b18b5b453e322d154bc55d5cc2754b0dfb4b2”
        - **content-length**
            - Beschreibung: Größe der Anhangsdatei in Byte
            - Format: Integer 64 Bit
            - Limitierung: Minimallänge = 1 Byte
            - Beispiel: 1048576
    - Beispiel: [{ "filename": "Wahlschein.pdf", "sha512sum": "f3b3ab3e6351e25b5c1...", "contentLength": 1048576 }]
- **stork_qaa_level (*)**
    - Beschreibung: Vertrauensniveau der Nachricht

| Technische Bezeichnung | Schlüssel | Bezeichnung nach TR-03160-1 bzw. Beschluss der Projektgruppe eID Strategie von Juli 2021 |
|---|---|---|
| STORK-QAA-Level-1 | 1 | Basisregistrierung |
| STORK-QAA-Level-2 | 2 | Niedrig |
| STORK-QAA-Level-3 | 3 | Substantiell |
| STORK-QAA-Level-4 | 4 | Hoch |

- Format: Integer 32 Bit
- Limitierung: Ein Wert aus dem Wertebereich = 1, 2, 3, 4
- Beispiel: 2

#### 4.2.1.4 Überprüfung des Nachrichteninhalts auf HTML-Codes

Alle vom ZBP empfangenen Nachrichten werden inhaltlich auf enthaltenen HTML-Code untersucht. Das ZBP lehnt Nachrichten auf Basis der unten genannten HTML-Richtlinie ab, die unerlaubte HTML-Tags und HTML-Attribute enthalten.

Die Felder, die dieser HTML-Prüfung unterzogen werden, sind im Folgenden:

- Absender (sender)
- Titel (title)
- Nachrichtentext (content)
- Service (service )
- Referenz / Aktenzeichen (reference)

Werden nicht erlaubte HTML-Tags im Nachrichteninhalt identifiziert, erhält der Sender den Error-Response "ZBP_400_004: HTML contains forbidden tags or attributes." zurück.

#### 4.2.1.5 HTML-Richtlinie

Diese Richtlinie beschreibt die HTML-Tags und -Attribute, die innerhalb der Anwendung zulässig sind. Jedes HTML-Tag und jedes Attribut wird mit seinen zulässigen Verwendungen aufgelistet und bietet so einen klaren Rahmen für die sichere Erstellung von Inhalten. Für die Werte und Inhalte der jeweiligen Attribute wird eine den Attributen gewöhnliche Ausprägung unterstellt und mithilfe von regulären Ausdrücken validiert, die in der HTML-Richtlinien XML-Datei definiert wurden. Somit kann hier keine erschöpfende Positivliste auf erlaubte Attributs-Inhalte dargestellt werden. Für Anpassungen an der HTML-Richtlinie muss sich an den Applikationsverantwortlichen gewendet werden.

**Zulässige HTML-Tags**

- <a>, <button>, <td>, <form>, <tr>, <table>, , , <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <br>, <b>, <style>, <title>, <p>, <html>, <head>, , , <body>, <meta>

**Zulässige Attribute**

- rel, charset, width,name, http-equiv, href, action, lang, type, style, target, cellspacing, cellpadding, class, align, bgcolor, role, content, border, method

**Verwendung von Attributen**

- **href**: kann verwendet werden von: <a>.
  - Beispiel für <a>: <a href="http://www.beispiel.com">Beispiellink</a>
- **style**: kann verwendet werden von: <h3>, <head>, <p>, <tr>, .
  - Beispiel für <h3>: <h3 style="color:blue;">Blauer Titel</h3>
- **lang**: kann verwendet werden von: <html>.
- **charset**: kann verwendet werden von: <meta>. Beispiel: <meta charset="UTF-8">
- **name** & **content**: können verwendet werden von: <meta>.
  - Beispiel für Metatags: <meta name="description" content="Beispielbeschreibung.">
- **http-equiv**: kann verwendet werden von: <meta>.
  - Beispiel: <meta http-equiv="X-UA-Compatible" content="IE=edge">
- **rel**: kann verwendet werden von: <link>, <a>.
  - Beispiel für <link>: <link rel="stylesheet" href="style.css">
  - Beispiel für <a>: <a href="http://www.beispiel.com" rel="noopener noreferrer">In neuem Tab öffnen</a>

- **role**: kann verwendet werden von: <tr>.
    - Beispiel für <tr>: <tr role="row">
- **width**: kann verwendet werden von: <table>, <tr>.
    - Beispiel für <table>: <table width="100%">
- **bgcolor**: kann verwendet werden von: <tr>.
    - Beispiel für <tr>: <tr bgcolor="#ff0000">
- **align**: kann verwendet werden von: <td>, <tr>.
    - Beispiel für <td>: <td align="center">Zentrierter Text</td>
- **target**: kann verwendet werden von: <a>.
    - Beispiel für <a>: <a href="http://www.beispiel.com" target="_blank" rel="noopener noreferrer">In neuem Tab öffnen</a>
- **class:** kann verwendet werden von: , <h3>, <tr>.
    - Beispiel für : Gestalteter Absatz.
- **method** & **action**: können verwendet werden von: <form>.
    - Beispiel für <form>: <form action="/submit-form" method="post"></form>
- **type**: kann verwendet werden von: <input>. Beispiel für <input>: <input type="text" name="firstname">

### 4.2.2 Antragstatusmeldung einstellen

Um eine Statusmeldung zu einem Antrag in das ZBP einzustellen, muss der Sender ein gültiges PKI-Zertifikat sowie den entsprechenden privaten Schlüssel besitzen. Beide dienen sowohl der Authentifizierung des Senders als auch der Signierung der Statusmeldung.

Die Identifikation des spezifischen Antrags eines Bürgers erfolgt über die eindeutige Kennung „applicationId“. Diese Kennung wird in den Statusmeldung-Inhalt integriert, sodass sie Teil der digitalen Signatur der Statusmeldung wird.

Die Statusmeldung-Schnittstelle selbst arbeitet synchron. Das heißt, dass der Empfang und die Prüfung der Signatur direkt durchgeführt werden. Antwortet die Schnittstelle mit HTTP-Statuscode 200 (OK), gilt die Statusmeldung als empfangen und geprüft.

Für neue Statusmeldungen werden jeweils Eingangsbenachrichtigungs-E-Mails über das Nutzerkonto an den betreffenden Bürger ausgelöst. Das ZBP verschickt selbst keine Eingangsbenachrichtigungen an Bürger. Das ist die Aufgabe des Nutzerkontos.

Es ist möglich eine Statusmeldung zu schicken, ohne den Status des Antrags zu verändern. Das ist bspw. dann der Fall, wenn die Behörde dem Bürger weiterführende Informationen mitteilen möchte, ohne den Status des Antrags selbst zu ändern. Dazu kann eine neue Statusmeldung gesendet werden, die den gleichen Statuswert, aber einen anderen Text als die vorherige Meldung hat.

Alle empfangenen Statusmeldungen werden inhaltlich auf enthaltenes HTML geprüft. Das ZBP lehnt Statusmeldungen ab, die HTML in einem der Felder enthalten. Wird HTML in einem der Felder identifiziert, erhält der Sender den folgenden Fehler-Response:

ZBP_400_001 "Value of the field 'de' in 'StatusDetailsDTO' is invalid (The value contains HTML)."

#### 4.2.2.1. Statusmeldung einstellen über Brückenkopf

Das ZBP ist ein REST-Service mit JSON als Format der Statusmeldung. Alternativ können Fachdienste und Fachverfahren Nachrichten an Bürger auch in anderen Protokollen (bspw. SOAP, OSCI) und Formaten senden. Dafür müssen Statusmeldungen in das ZBP über einen sogenannten Brückenkopf als Übersetzer eingestellt werden. Der Brückenkopf empfängt dazu die Statusmeldung im Format des Fachdienstes oder -verfahrens, übersetzt sie in das vom ZBP erwartete Format und sendet sie dann weiter an das ZBP. Da das ZBP Anfragen (Requests) synchron beantwortet, erhält der Brückenkopf direkt die Antwort (Response) vom ZBP und kann diese wiederum in das Format des entsprechenden Fachdienstes oder -verfahrens übersetzen und an sie weitergeben.

In dieser Statusmeldung-Transportkette übernimmt der Brückenkopf die Rolle des Senders der Statusmeldung und der Fachdienst oder das Fachverfahren weiterhin die Rolle des Autors der Statusmeldung (Statusmeldung-Ersteller). Beide Rollen müssen jeweils über ein gültiges PKI-Zertifikat und den entsprechenden privaten Schlüssel verfügen, um sich gegenüber dem ZBP zu authentifizieren und um den Inhalt der Statusmeldung zu signieren.
Für eine Anfrage an das ZBP muss der Brückenkopf als Sender, wie im Abschnitt JSON Web Token (JWT) erläutert, den Bearer-Token-String als Wert für den Request-Header “Authorization” setzen, sein PKI-Zertifikat im PEM-Format für den SSL- / TLS-Verbindungsaufbau angeben, mit seinem privaten Schlüssel der PKI die Signatur der Statusmeldung vornehmen und über das Feld “sha512sum” im Statusmeldung-Objekt übergeben.
Der Fachdienst / das Fachverfahren muss als Autor der Statusmeldung gegenüber dem ZBP beweisen, dass er ein valider Autor ist. Dazu muss dieser sein JSON Web Token (JWT) in Form eines Token-String und sein PKI Client-Zertifikat mit enthaltenen öffentlichen Schlüssel (Public Key) dem Brückenkopf in seiner Anfrage mitteilen. Der Brückenkopf wiederum muss diese Werte als Sender der Statusmeldung

über die Felder “authorToken” und “authorCertificate” im Statusmeldung-Objekt an das ZBP weitergeben.

Details zur Nutzung eines Brückenkopfes müssen individuell mit dem jeweiligen Brückenkopf geklärt werden. Jeder Brückenkopf kann eigene Anforderungen an die Statusmeldung definieren.

#### 4.2.2.2 Mögliche Statuswerte für eine Statusmeldung

| Statuswert in der Antragsfortschritt-anzeige | String | Verpflich-tend zu setzen | Zu setzen von | Erläuterung |
|---|---|---|---|---|
| Antrag in Erstellung | INITIATED | ✗ | Online-dienst | Gedacht für Onlinedienste, die Anträge zwischenspeichern können. Der begonnene Antrag ist für den Bürger bereits nach dem Senden dieses Status in der Antragsliste zu sehen. |
| Antrag versendet | SUBMITTED | ✗ | Online-dienst | Gedacht für Onlinedienste: Sollte der Antrag vom Onlinedienst asynchron an das Fachverfahren übergeben werden oder der Antrag aus unbekanntem Grund nicht im Fachverfahren eingehen, kann man so dennoch erkennen, dass der Bürger den Antrag erfolgreich gesendet hat. |
| Antrag eingegangen | RECEIVED | ✓ | Fachverfah-ren | Obligatorischer Status, der vom empfangenen Fachverfahren gesetzt werden soll, sobald der neue Antrag eingegangen ist. |

| | | | | |
|---|---|---|---|---|
| Antrag in Prüfung | PROCESSING |  | Fachverfahren | Der Bürger wird beim Antragsstatus "Eingegangen" annehmen, dass sein Antrag auch irgendwann bearbeitet wird. Eine Bestätigung, dass sich jetzt konkret ein Sachbearbeiter um den Antrag kümmert, ist jedoch ein großer Gewinn bzgl. Komfort und Akzeptanz der digitalen Behördenleistungen. Daher wird empfohlen diesen Status zu verwenden. |
| Aktivität erforderlich | ACTION_REQUIRED |  | Fachverfahren | Das Fachverfahren sollte dem Bürger immer eine Nachricht im ZBP hinterlegen (inkl. der damit verbundenen E-Mail-Benachrichtigung), wenn eine Aktivität seinerseits notwendig ist. Eine zusätzliche Visualisierung am Antrag selbst kann dies aber noch deutlicher hervorheben. Des Weiteren wird der Bürger damit auch daran erinnert, dass eine Aktivität seinerseits ggf. immer noch aussteht.<br><br>Nach erfolgreicher Aktivität des Bürgers (bspw. Nachreichen von Dokumenten), sollte der Status wieder zurück auf "Antrag in Prüfung" gestellt werden. |
| Antrag abgeschlossen | COMPLETED |  | Fachverfahren | Der Status "Abgeschlossen" steht für die Beendigung des Antrags. Er stellt keine Aussage drüber dar, ob der Antrag des Bürgers "erfolgreich" war. |

#### 4.2.2.3 Das Statusmeldung-Objekt

Eine Statusmeldung ist ein Objekt im JSON-Format, das aus den nachfolgend beschriebenen Feldern besteht. Die zwingend erforderlichen Pflichtfelder sind mit **(*)** gekennzeichnet:

- **content (*)**
    - Beschreibung: Ist der eigentliche Inhaltsteil der Statusmeldung. Der Wert muss in Form eines JSON-Strings über alle angegebenen Inhaltsfelder (optionale und Pflichtfelder) gebildet werden.
    - Format: JSON-String
    - Beispiel:
      "{\"applicationId\":\"e8bc5ebc-56cf-4ff0-a861-f989aa1ba4a9\",\"status\":\"PROCESSING\",\"publicServiceName\":{\"de\":\"Wahlschein-Service\"},\"statusDetails\":{\"de\":\"Antrag wird bearbeitet\"},\"additionalInformation\":{\"de\":\"Ihr Antrag wird von einem Sachbearbeiter bearbeitet.\"},\"reference\":\"AZ123456\",\"senderName\":\"Bundeswahlscheinverfahren\",\"createdDate\":\"2023-11-01T11:23:21.223Z\"}"
- **sha512sum (*)**
    - Beschreibung: Ist die Signatur der Statusmeldung. Der Wert muss mittels dem privaten Schlüssel des PKI-Zertifikats und mit dem RSA512 Algorithmus über den JSON-String des “content”-Felds gebildet werden.
    - Format: String
    - Beispiel:
      "hWa2YSEHza+EHh2wL4G7t5nDlHMmcRNIqQYRG-bRA9wVeG+r6cmSSiWQ+/p8DNjZmEN3Cq2cC2YGfRKbLtnoHxaLKABio-aKj0sHGbBK407siZNTNXnEXkQW46cu1/g9KekX6Y+u5orUJ7ac0RDanYR-WoiZRmUurt/0kH4aCGtGVZivo9Q+vyVdgUuXqJp2JKFhD6D5XTEd0Ly90Q9ThILsaizOp-mRMcgp44x88TG3yA7GzTb5SRULz/XHDmaa64/qUw4qksJWxbwgRvw/JGXVLqflL-zwH+bDsfxspjbttUTxH8ItHh6G/Ok0FbRbrK-MiK3E76PCreon47GlXxKDnybis3wih8TFUqWm/vajOogmn9fTZWnc7f/aa2TMtyf-KRqgqNF+IQqQPBgC5ZBvwdeqnUX71fFsC5srd0ocWs6QR6OgXIzjt1RqqOp+Pfbe-TmwQPhEJQerzHCAfcS6moS0zolr5AfEHbExRWbmBHowS0I9YKF+Ti-

iFYqGu4aiQKzGFS1GKesDq7mnQU+rjqqCmuJls-
BcVk0pCg3iQCxgkXWdbbgxxaR8dtOejjrmABaYaH6nsv/1H2O47tct8FWDtExzThl9lAN+
KHeYhvCzyYAQNsFLSJK4eIvF7VqzgJYMXnBKS1XifICtpAc0Khk9RTT-
vxR1dHvZvO5RIZPsEp9hnA="

- **authorToken**
    - Beschreibung: Ist das JSON Web Token (JWT) in Form eines Token-String. Dieser Wert muss angegeben werden, wenn die Statusmeldung über einen Brückenkopf an das ZBP gesendet wird und somit der Sender und Signierer der Statusmeldung nur der Übermittler (Brückenkopf) und nicht der eigentliche Autor der Statusmeldung (Drittanwendung) ist.
    - Hinweis: Dieses Feld steht in Abhängigkeit zum Feld “authorCertificate”. Das bedeutet, dass immer auch ein valider Wert für “authorCertificate” angeben werden muss, sobald ein Wert für “authorToken” übermittelt wird. Dies ist notwendig, damit überprüft und sichergestellt werden kann, dass der Autor der Statusmeldung auch über den privaten Schlüssel (Private Key) des PKI-Zertifikats verfügt.
    - Format: String
    - Beispiel: “eyJhbGciOiJSUzUxMilsIn….”
- **authorCertificate**
    - Beschreibung: Ist das PKI-Zertifikat des Autors als PEM-Format String. Dieser Wert muss angegeben werden, wenn die Statusmeldung über einen Brückenkopf an das ZBP gesendet wird und somit der Sender und Signierer der Statusmeldung nur der Übermittler (Brückenkopf) und nicht der eigentliche Autor der Statusmeldung (Drittanwendung) ist.
    - Hinweis: Dieses Feld steht in Abhängigkeit zum Feld “authorToken”. Das bedeutet, dass immer auch ein valider Wert für “authorToken” angeben werden muss, sobald ein Wert für “authorCertificate” übermittelt wird. Dies ist notwendig, damit überprüft und sichergestellt werden kann, dass der Autor der Statusmeldung auch über den privaten Schlüssel (Private Key) der PKI-Zertifikats verfügt.
    - Format: String

- Beispiel: "-----BEGIN CERTIFICATE-----MII-FIjCCAwoCFBRs6KrFt2pgOK9G750NZZ6aQ6v+MA[…]B0rdG84mEM9oQ==-----END CERTIFICATE-----"

#### 4.2.2.3 Der Statusmeldungsinhalt

Eine Statusmeldung unterstützt im Inhaltsteil folgende fachliche Parameter, wobei zwingend erforderliche Pflichtfelder mit **(*)** gekennzeichnet sind:

- **applicationId (*)**
  - Beschreibung: Ist die eineindeutige Antrags-ID und dient für die Identifizierung des Bürgerpostfachs
  - Format: UUID-String
  - Beispiel: "applicationId":"e8bc5ebc-56cf-4ff0-a861-f989aa1ba4a9"
- **status (*)**
  - Beschreibung: Ist der Statuswert.
  - Format: String
  - Limitierung: Darf nur ein gültiger bzw. bekannter Statuswert sein (siehe Statuswerte).
  - Beispiel: "status":"PROCESSING"
- **publicServiceName (*)**
  - Beschreibung: Ist der Titel des Antrags (Leistung).
  - Format: Objekt mit verfügbarer Sprache / Übersetzung.
  - Limitierung: Jedes Objekt muss mindestens eine Sprachangabe enthalten. In der aktuellen ZBP-Version wird nur die Sprache Deutsch unterstützt und wird wie folgt angegeben:
    - **de**
      - Beschreibung: Deutschsprachiger Titel des Antrags
      - Format: String
      - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 100 Zeichen

- Beispiel: "publicServiceName":{ "de ": "Anmeldung Ihres Zweitwohnsitzes"}

- **senderName (*)**
  - Beschreibung: Ist der Name des Senders der Statusmeldung.
  - Format: Objekt mit verfügbarer Sprache / Übersetzung.
  - Limitierung: Jedes Objekt muss mindestens eine Sprachangabe enthalten. In der aktuellen ZBP-Version wird nur die Sprache Deutsch unterstützt und wird wie folgt angegeben:
    - **de**
      - Beschreibung: Deutschsprachiger Name des Senders der Statusmeldung.
      - Format: String
      - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 100 Zeichen
  - Beispiel: "additionalInformation":{"de":"Landeshauptstadt München"}
- **createdDate (*)**
  - Beschreibung: Ist das Datum der Erstellung der Statusmeldung durch den Sender.
  - Format: String als date-time
  - Beispiel: "createdDate":"2023-11-01T11:23:21.223Z"
- **statusDetails**
  - Beschreibung: Sind die Detailinformationen zum Status-Update.
  - Format: Objekt mit verfügbarer Sprache / Übersetzung.
  - Limitierung: Jedes Objekt muss mindestens eine Sprachangabe enthalten. In der aktuellen ZBP-Version wird nur die Sprache Deutsch unterstützt und wird wie folgt angegeben:
    - **de**
      - Beschreibung: Deutschsprachige Detailinformationen zum Status-Update.
      - Format: String

        - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 50 Zeichen
    - Beispiel: "statusDetails":{"de":"Antrag in Prüfung"}

- **additionalInformation**
    - Beschreibung: Sind die Zusatzinformationen zum Antrag hinsichtlich des Status-Update.
    - Format: Objekt mit verfügbarer Sprache / Übersetzung.
    - Limitierung: Jedes Objekt muss mindestens eine Sprachangabe enthalten. In der aktuellen ZBP-Version wird nur die Sprache Deutsch unterstützt und wird wie folgt angegeben:
        - **de**
            - Beschreibung: Deutschsprachige Zusatzinformationen zum Antrag hinsichtlich des Status-Update.
            - Format: String
            - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 100 Zeichen
    - Beispiel: "additionalInformation":{"de":"Ihr Antrag wird derzeit geprüft. Die voraussichtliche Bearbeitungszeit beträgt 4-6 Wochen."}

- **reference**
    - Beschreibung: Ist die Referenz / das Aktenzeichen zu einem Vorgang.
    - Format: String
    - Limitierung: Minimallänge = 1 Zeichen; Maximallänge = 50 Zeichen
    - Beispiel: "reference":"AZ-123456"

# 5. Fehlercodes

## 5.1 Fachliche Fehlercodes

Hier werden die fachlichen Fehlercodes der Schnittstellen des ZBP-Dienstes beschrieben.

| Fehlercode | Fehlertext | Beschreibung | Schnittstellen |
|---|---|---|---|
| ZBP_400_001 | Invalid request body. | Die zu übertragende Nachricht im Body der Anfrage fehlt oder ist unvollständig. | Nachricht einstellen, Antragstatusmeldung einstellen |
| | Value of the field '{field-name}' in '{dto-name}' is invalid ({reason}). | Wert des Feldes '{field-name}' in '{dto-name}' is invalide ({reason}). | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_400_002 | Multipart form is malformed. | Die Beschreibung der mehrteiligen Daten im Multistream-Format konnte nicht gelesen werden, da sie Fehler enthält. | Nachricht einstellen |
| ZBP_400_003 | Invalid attachment type : {attachment-type}. | Der Dateityp {attachment-type} des hochgeladenen Anhangs ist ungültig. | Nachricht einstellen |
| ZBP_400_004 | HTML contains forbidden tags or attributes. | Der Nachrichteninhalt enthält nicht erlaubte Tags oder Attribute. | Nachricht einstellen |
| ZBP_400_005 | Attachment {filename} missing in json content. | Ein hochgeladener Anhang ist nicht im JSON der Nachricht referenziert. | Nachricht einstellen |
| ZBP_400_006 | {field} missing for {filename} in Attachment in json content. | Ein Feld im Attachment-Teil der JSON-Nachricht fehlt. | Nachricht einstellen |

| | | | |
|---|---|---|---|
| ZBP_400_007 | Can't find {header-name} in Header. | Der Header-Name {header-name} wurde nicht im Header gefunden. | Nachricht einstellen |
| ZBP_400_008 | Duplicate filename in message: {filename}. | Die Nachricht enthält unerlaubt gleiche Dateianhänge. | Nachricht einstellen |
| ZBP_400_012 | Missing or incomplete message in body. | Die Validierung der Felder in einer Nachricht ist gescheitert. | Nachricht einstellen |
| ZBP_400_013 | Unable to create Envelope / DTO object from the body, Incorrect json content. | Das JSON der Nachricht ist technisch nicht korrekt und konnte deshalb nicht geparst werden. | Nachricht einstellen |
| ZBP_400_014 | Message content is too long, allowed max length: %s. | Der Nachrichtentext ist zu lang in Bezug auf die maximal erlaubte Zeichenlänge für Nachrichten, die in der ZBP-Konfiguration definiert ist. | Nachricht einstellen |
| ZBP_401_001 | Malformed authorization token. | Der Autorisierungs-Token für die spezifische Anfrage ist ungültig.<br>Mögliche Ursachen: Der Token hat eine ungültige Form oder enthält unzureichende Berechtigungen für den Zugriff auf die angeforderte Ressource. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_401_002 | Client token could not be validated. | Das Zertifikat für die spezifische Anfrage ist ungültig.<br>Mögliche Ursachen: Das Zertifikat stammt nicht von der PKI oder das Zertifikat wurde von der PKI zurückgezogen. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_403_001 | Access Denied. | Der Token hat keine Berechtigung diese Ressource zu benutzen. | Nachricht einstellen, Antragstatusmeldung einstellen |

| | | | |
|---|---|---|---|
| ZBP_403_002 | Given signature does not match with message content. Please re-sign and try again. | Die übertragene Prüfsumme stimmt nicht mit dem Nachrichteninhalt überein. | Nachricht einstellen |
| ZBP_404_003 | Can't find this mailbox. | Das Postfach konnte nicht gefunden werden. | Nachricht einstellen |
| ZBP_404_008 | Can't find this application. | Der Antrag konnte nicht gefunden werden. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_409_001 | No trust level. | Es wurde kein Trustlevel angegeben. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_409_002 | No subject in token. | Es wurde kein Subject (Nutzerkonto-ID) im Token angegeben. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_409_003 | Wrong trust level. | Der angegebene Trustlevel ist ungültig. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_409_007 | Invalid characters in application reference number. | Die Zeichenfolge der Fallreferenz-ID enthält eine nicht unterstützte Zeichencodierung. | Nachricht einstellen |
| ZBP_409_008 | Application reference number is too long. | Die Zeichenfolge der Fallreferenz-ID hat mehr als 255 Zeichen. | Nachricht einstellen |
| ZBP_409_009 | This state transition is not allowed. | Der übertragene Status ist zwar prinzipiell gültig, der Wechsel zu diesem Status ist aber nicht erlaubt. | Antragstatusmeldung einstellen |

| | | | |
|---|---|---|---|
| ZBP_413_001 | Number of allowed attachments exceeded. | Die Anzahl der erlaubten Anhänge wurde überschritten. | Nachricht einstellen |
| ZBP_413_002 | Sum of attachments size exceeded limit. | Die erlaubte Summe der Dateigrößen der Anhänge wurde überschritten. Es wird hier die Gesamtgröße aller Anhänge zusammengerechnet. | Nachricht einstellen |
| ZBP_500_006 | Error processing request. | Beim Verarbeiten der Anfrage ist ein Fehler aufgetreten. | Nachricht einstellen |
| ZBP_500_007 | Error validating the message content. | Beim Überprüfen des Nachrichteninhalts auf unerlaubtes HTML ist ein interner Fehler aufgetreten | Nachricht einstellen |
| ZBP_500_011 | Internal server error occurred. | Ein interner Serverfehler ist aufgetreten. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_500_012 | Please check logs for more information. | Bitte überprüfen Sie die Logs für weitere Informationen. | Nachricht einstellen, Antragstatusmeldung einstellen |
| ZBP_503_001 | Error uploading to file storage. | Beim Hochladen der Datei zum Dateiserver ist ein Fehler aufgetreten. | Nachricht einstellen |

## 5.2 Andere Fehlercodes

Bei der Kommunikation mit dem ZBP können Fehlercodes von dem ZBP vorgelagerten System, wie z. B. einem Proxy-Server, zurückgegeben werden.

Ein Beispiel für einen nichtfachlichen Fehlercode kann in den Beispielen eingesehen werden.

## 5.3 Fehlercodes-Zuordnung (Mapping)

Das Nutzerkonto und die Drittanwendungen müssen die ZBP-Fehler abfangen und die ZBP-Fehlertexte jeweils in benutzerfreundliche Fehlertexte übersetzen (mapping), um diese dann in der GUI hinsichtlich der gewählten Sprache (Deutsch, Englisch, Russisch, Ukrainisch) anzuzeigen.

# 6. Beispiele

## 6.1 Beispiele zur Erstellung des JSON Web Token

### 6.1.1. Beispiel: Inhalt des JSON Web Token

**Header: Algorithmus & Token-Typ**

```
{
  "alg": "RS512",
  "typ": "JWT"
}
```

**Payload: Daten**

```
{
  "iat": 1715760458,
  "exp": 1715762258,
  "signer": "Testclient",
  "roles": [
    "THIRD_PARTY"
  ]
}
```

**Signatur**

```
RSASHA512(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  {private_key}
)
```

### 6.1.2 Beispiel: Erstellung JSON Web Token in Java

```
JWT.create()
  .withIssuedAt(createdDate) // createdDate = Erstellungsdatum
  .withExpiresAt(expiredDate) // expiredDate = Ablaufdatum
  .withClaim("signer", signer) // signer = CN (= common name) des PKI-
Zertifikats
  .withClaim("roles", List.of("THIRD_PARTY")) // roles = Rolle
  .sign(Algorithm.RSA512(privateKey)); // Signierung des JWT
```

### 6.1.3 Beispiel: Erstellung JSON Web Token per bash-Skript

```
#!/bin/bash

privateKey="path_to_your_private_key.pem"
issuedAt=$(date +%s)
expiresAt=$(date --date='+1 hour' +%s)
signer="CN_of_PKI_Certificate"
roles="THIRD_PARTY"

header=$(echo -n '{"typ":"JWT","alg":"RS512"}' | openssl base64 -e -A | tr '+/' '-_' | tr -d '=')

payload=$(echo -n "{\"iat\":$issuedAt,\"exp\":$expire-
sAt,\"signer\":\"$signer\",\"roles\":[\"$roles\"]}" | openssl base64 -e -A | tr '+/' '-_' | tr -d '=')

signature=$(echo -n "$header.$payload" | openssl dgst -sha512 -sign $privateKey | openssl
base64 -e -A | tr '+/' '-_' | tr -d '=')
```

```
# Print Full JWT
echo "$header.$payload.$signature"
```

## 6.2 Beispiele zum Einstellen von Nachrichten

### 6.2.1 Beispiel: Einstellen einer Nachricht mit Anhang in Java

#### 6.2.1.1 Berechnung der Checksumme der Anhänge:

```
String getSha512HexOfInputStream(InputStream input) {
   return org.apache.commons.codec.digest.DigestUtils.sha512Hex(input);
}
```

#### 6.2.1.2 Beispiel Nachrichteninhalt als JSON

```
{
   "mailboxUuid": "0f0407c5-7f7d-4ada-8dfe-43760d90586d",
   "stork_qaa_level": 1,
   "sender": "Bundeswahlscheinverfahren",
   "title": "Nachrichtenbetreff vom 1.8.2023, 23:08:05",
   "content": "Nachrichtentext vom 1.8.2023, 23:08:05",
   "service": "Wahlschein-Service",
   "retrievalConfirmationAddress": "info@example.com",
   "replyAddress": "info@bund.de",
   "attachments":
   [
      {
         "filename": "Wahlschein.pdf",
```

```
            "sha512sum": "f3b3ab3e6351e25b5c1...",
            "contentLength": 13264
        }
    ],
    "reference": "80364006",
    "senderUrl": "https://www.bund.de",
    "applicationId": "e8bc5ebc-56cf-4ff0-a861-f989aa1ba4a9"
}
```

#### 6.2.1.3 Signierung des Nachrichteninhalts

```
String signString(java.security.interfaces.RSAPrivateKey privateKey, String input)
        throws java.security.InvalidKeyException, java.security.SignatureException,
        java.security.NoSuchAlgorithmException {
    java.security.Signature signature = java.security.Signature.getInstance("SHA512withRSA");
    signature.initSign(privateKey);
    signature.update(input.getBytes(java.nio.charset.StandardCharsets.UTF_8));
    byte[] signed = signature.sign();
    return java.util.Base64.getEncoder().encodeToString(signed);
}
```

#### 6.2.1.4 Erstellung der Nachricht und senden als Multipart-Request

```
    private org.springframework.http.ResponseEntity<de.akdb.egov.zbp.dto.CreateMessageResponseDTO> sendMessageEnvelope(
            org.springframework.web.client.RestTemplate restTemplate,
            com.fasterxml.jackson.databind.ObjectMapper objectMapper,
            java.net.URI putMessageUri,
```

```
        de.akdb.egov.zbp.dto.CreateMessageV6DTO dto,
        java.util.List<org.springframework.core.io.FileSystemResource> resources,
        java.security.interfaces.RSAPrivateKey privateKey,
        java.lang.String thirdPartyToken
    ) throws com.fasterxml.jackson.core.JsonProcessingException,
        java.security.InvalidKeyException,
        java.security.SignatureException,
        java.security.NoSuchAlgorithmException
    {
        org.springframework.util.MultiValueMap<String, Object> body = new org.springframework.util.LinkedMultiValueMap<>();
        java.lang.String value = objectMapper.writeValueAsString(dto);
        java.lang.String signString = signString(privateKey, value);
        java.lang.String authorToken = null; // only filled for message brokers (Brückenköpfe)
        java.lang.String authorCertificate = null; // only filled for message brokers (Brückenköpfe)
        de.akdb.egov.zbp.dto.CreateMessageEnvelopeV6DTO signedEnvelope = new de.akdb.egov.zbp.dto.CreateMessageEnvelopeV6DTO(value, signString, authorToken,
            authorCertificate);

        body.add("json", objectMapper.writeValueAsString(signedEnvelope));
        resources.forEach(resource -> body.add("files", resource));

        org.springframework.http.HttpHeaders headers = new HttpHeaders();
        headers.setContentType(org.springframework.http.MediaType.MULTIPART_FORM_DATA);
        headers.setBearerAuth(thirdPartyToken);

        return restTemplate.exchange(putMessageUri, org.springframework.http.HttpMethod.PUT,
```

```
        new HttpEntity<>(body, headers), de.akdb.egov.zbp.dto.CreateMessageResponseDTO.class);
    }
```

### 6.2.2 Beispiel: Einstellen einer Nachricht mit Anhang mit cURL

```
# Checksummen aller Dateianhänge berechnen:
sha512sum <file-path>

# Dateigröße (contentLength) aller Dateianhänge ermitteln
ls -l <file-path>

# Checksumme des Nachrichteninhalts-JSONs (content) inkl. des Dateianhang-Arrays (attachments), die mit oben ermittelten Daten gefüllt wird
printf '{"mailboxUuid":"<uuid>","stork_qaa_level": ,"sender":"","title":"","content":"","service":"","retrievalConfirmationAddress":"","replyAddress":"","attachments":[{"filename":"<filename>","sha512sum":"<value-from-sha512sum-command>","contentLength":<value-from-ls-command>}],"reference":"","senderUrl":"","applicationId":""}' | openssl dgst -sha512 -sign <private-key-path> -out msg-digest.txt

# base64-Kodierung der Checksumme
base64 -w 0 msg-digest.txt

# Aufbau des Nachrichten-Objekts inkl. der Checksumme: Doppelte Anführungsstriche für Strings müssen mit einem \ "escaped" werden
curl -k -X 'PUT' \
 'https://<server-host>/v6/mailbox/messages' \
 -H 'accept: application/json' \
 -H 'Authorization: Bearer '<jwt-token>' \
```

```
-H 'Content-Type: multipart/form-data' \
-F 'json={"content":"{\"mailboxUuid\":\"\",\"stork_qaa_level\":,\"sender\":\"\",\"title\":\"\",\"content\":\"\",\"service\":\"\",\"retrievalConfirmationAddress\":\"\",\"replyAddress\":\"\",\"attachments\":[{\"filename\":\"\",\"sha512sum\":\"\",\"contentLength\":}],\"reference\":\"\",\"senderUrl\":\"\",\"applicationId\":\"\"}",
"sha512sum":"<output-from-base64-command>"}' \
-F 'files=<file-path>@;type=<file-type>'
```

### 6.2.3 Beispiel: HTTP-Kommunikation beim Einstellen einer Nachricht ohne Anhänge

#### 6.2.3.1 Request

```
PUT /v6/mailbox/messages HTTP/2
Host: zbp-pre-a-bund.akdb.de
User-Agent: curl/8.4.0
accept: application/json
Authorization: Bearer eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJpY
XQiOjE3MTU3NzIwMjAsImV4cCI6MTcxNTc3MzgyMCwic2lnbmVyIjoiQmF5ZXJpc
2NoZV9FaW53b2huZXJhbmdlbGVnZW5oZWl0ZW4iLCJyb2xlcyI6WyJUSElSRF9QQ
VJUWSJdfQ.MQO4cfYdJAYQy0h3Jm-1tSivsCsmjmBS91VhlrE-6DpDjIg1FakA8e
qYWE4Pf4igiTTm5js6nYw7w4gg5kF-vQGDQo-ZrHKdnb6wL7T_A_mLOllIfMrdyb
K0pkWD7jRqfhfw9sTGfqHxU6L5S0lRFinnOGukschbgcQGpeaejAG4L8Aelu7cGf
C65s-0fmyxhGe7nQostCDnqkLAs26GAbamb0rywT-wSD_GA_c9Iw766mgWRVVl7d
Q9gvNwiXTr6LLHxjbUw3WAWy_i8q-KZO2-M03Nf81ru42BC8rU3BzDMg8Kh5bLkN
JUHuhzm7CEWsUb_MhIdmoJfVMpzdzoAABqFRFm9Mppn9J5Z-rDdzbFIrSHE4F6vv
8icTV-VWHpzf_NPKTgs6lNXjxU6qd_HuubBkwIbMCmjjepnoESmasit0NP_lb4_j
2quBfprRpNsafDENEz_OU_Zoyo80jkTeVUSfVZsRgnTCG4zuw6-kDTseb0gfwFup
YfLLYoQ5ZYQvQjJn4VDNtD5-Bea_yD1qGHC_aGrAzOKzuW4_X4c3WljNT4kJViHQ
G-g7toSVHyrgzqT3LXn6kvSXbP0-c8y7xlsqYeY7xRMlAOajdE1Bx89g10tRD_Rz
```

Q-Ps9g2opSbOyaSxzwUPE-BB33yGkck1CpfCyB85ClWxiyDjGvxFE

Content-Length: 1338

Content-Type: multipart/form-data; boundary=----------

----nnDWNm1Mt0rCIqmIha5Kqn

----------nnDWNm1Mt0rCIqmIha5Kqn

Content-Disposition: form-data; name="json"

{"content":"{\"mailboxUuid\":\"45d366d6-775c-4b46-8128-039866e17
608\",\"stork_qaa_level\":1,\"sender\":\"Bayerische_Einwohnerang
elegenheiten\",\"title\":\"Nachricht ohne Anh..nge\",\"content\"
:\"Nachrichtentext vom 10.5.2024, 13:23:25\",\"service\":\"Servi
ce\",\"retrievalConfirmationAddress\":\"mail@example.com\",\"rep
lyAddress\":\"reply@example.com\",\"attachments\":[],\"reference
\":\"95245912\",\"senderUrl\":\"www.example.com\",\"caseId\":\"1
ac1bffc-310d-4cf7-8c1c-772c0c9c9082\"}","sha512sum":"PUAKRxTayO5
tC4nZ5ecbY9ql1YM5TXueQVA0L89zSkv0UG5Ybf8+Gax4Oa4SftI/+GHeZdvk7Pw
WXy7yyZiI1345MH13VDhahdLyDQMH9I6uO/mw+sGHxRdLCE7HK62kzqyV/caPFW0
wnpel/3YUEyKjbp0jpUl3vdY+lfcdiirAZxWNFsuK/MIAjgJjEdAx60kP+qVva7f
YFxj70T8Y75x1Z6W5tsP2kAhsEYERaOMIH9TBFEGol4KhMpBYaslwPhUU0pmmNB7
HhIPfSr15FKFq+YeubRiQ8f07wTN4KEHP+3RcvyRsKIyT7XG05B3RsjlQMX8vPF9
lhnw2bAlPrVESyK/PljJZzviywOiYrDjt234vxlt0u6mqxcwbJSHcJtoQi1knLbk
tbZkTBU7o71bXLKB1erVx5tOGzR/Kb4UG3XSnDOsiq9ZJfsX0iYIC366BHRADv9A
KxMcOp8Cnv3GLK5cPeWtpegtADjPHqglxugqCRGeaoANzr640dz2alf7+mD/kNN8
fVXLJVMZddChFlpFXArR1Eo80o9zQE9gQmIaZmvaVaaFQUDpy98Iw2SloKR2MEdM
hri2ykxQLeBmto3DZ46KzFvmb2O4Gb/BMDCjkDdz+nK1bXX2BivQWlRGg0Y7U4N5
npmPHa/oEeOmNxrREtfAXcPRuD3jJ6yY="}

```
--------------------------nnDWNm1Mt0rCIqmIha5Kqn--
```

#### 6.2.3.2 Response

```
HTTP/2 200
date: Wed, 15 May 2024 11:43:37 GMT
content-type: application/json
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
cache-control: no-cache, no-store, max-age=0, must-revalidate
pragma: no-cache
expires: 0
strict-transport-security: max-age=31536000; includeSubDomains; preload
x-robots-tag: noindex, nofollow
server: Apache

{"mailboxHandle":"45d366d6-775c-4b46-8128-039866e17608","messa-
geId":1715773392120,"messageUuid":"843ac6c3-7175-44e6-83f2-9d3db5bcfebe"}
```

### 6.2.4 Beispiel: HTTP-Kommunikation beim Einstellen einer Nachricht mit mehreren Anhängen

#### 6.2.4.1 Request

```
PUT /v6/mailbox/messages HTTP/2
Host: zbp-pre-a-bund.akdb.de
```

```
User-Agent: curl/8.4.0
accept: application/json
Authorization: Bearer eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJpY
XQiOjE3MTU3NjU3OTMsImV4cCI6MTcxNTc2NzU5Mywic2lnbmVyIjoiQmF5ZXJpc
2NoZV9FaW53b2huZXJhbmdlbGVnZW5oZWl0ZW4iLCJyb2xlcyI6WyJUSElSRF9QQ
VJUWSJdfQ.R62NZP8_Z0CkdC-Dhv5C2xHUX3o4wnrfdOAe-SrLPloKzWSER7cb91
2_RcFjtIqiU9Cbrgg-KBJwj4s0GjZ6apdKBztVDerfKcDzl4LsKFELtlLKXqHXrD
aWKjaWyCSuIN3a9reMC0QoIqQKKJ-fUYFRzyGGACLcXPtVkEpwyhxRabNc0vOv_H
q56FwyP5WQ-OAJ4vhI2vuyo0xQVee84evnrtwlhM6758VKa_VaXP6Q0Ddpz3MHSL
UMb9xrZj_ZXYSw2bVXxhwdFp02ei9etkOs9APZrbQuRKMBxekyuILoRU2A73G2V_
7efII66Ig7MnAth0TyYSeYiTp9EGcXYVoRLlodFx-AT4vKQg2P-reWua4XZuDKgJ
iY8ckGokvyQkdnDAqOKrclx4z2Z-3o3lKo8U3XLJxojBDgE7WvfhsjsJQJWJAT1S
PWw_ancYrB0p_gQFiS3f1hM05oWzUFZp63DqnNVHEMVNHdW6ZGgk-2A92ADRV8ei
IeIsgi5JIVgZ1iX6BNNj3KJvtpPlMr6n49ij8LoFJIe7NpQ4k6FhoOGqd5mM1KP9
qybrbCEtdGtYLhbycgAECLud3WR46chJtG7PLHH97GKrn9Vu6fa6bIH9tHDmkeMR
XA9zYbiMcl9qEutogaGL08ZyjObqKZv2J4Q7y6d8-_OMUMRhqyi5o
Content-Length: 2151
Content-Type: multipart/form-data; boundary=--------------------
----ZYrENGBynEosq434dQEpDS

--------------------------ZYrENGBynEosq434dQEpDS
Content-Disposition: form-data; name="json"

{"content":"{\"mailboxUuid\":\"45d366d6-775c-4b46-8128-039866e17
608\",\"stork_qaa_level\":1,\"sender\":\"Bayerische_Einwohnerang
elegenheiten\",\"title\":\"Nachrichtenbetreff vom 10.5.2024, 11:
30:27\",\"content\":\"Nachrichtentext vom 10.5.2024, 11:30:27\",
```

\"service\":\"Service\",\"retrievalConfirmationAddress\":\"\",\"
replyAddress\":\"\",\"attachments\":[{\"filename\":\"Textdatei.t
xt\",\"sha512sum\":\"7bce225fbab618e2e804e53cb7d16ebb29ea2d36864
46a20760e04923bcd6b7b6ab87d647b2551e46c75820d884d5c7801601c3b5c3
47afa5594c8667b8c2c5e\",\"contentLength\":34},{\"filename\":\"Do
kument.pdf\",\"sha512sum\":\"2ee8d9fbe74a1794140a5199758f9c3b3aa
55f883686f73b848c7615ecc11b0671a5f28d200d61a1a7361da91fabd4b72a4
a6af420852d79ca0a44ab5fdbac53\",\"contentLength\":134}],\"refere
nce\":\"514743618\",\"senderUrl\":\"\",\"caseId\":null}","sha512
sum":"XRHoyUfEet391r3U4EaNaJr1+1/wUj3LOqvpYAaM3c5GSbALukSCjXr2uR
g/dLC6iqzp4VTlVXxwzCGfTPoEcD+4DmLD3RUlc8d7CN3z8Ap7EKpzzxm/xQ7ASK
HW7oeQWvy40KBWSLeU8AwIzM/RZHweis/LXqmeZAYLhT/qKKTeVhmNIxLqJPXUKM
bxbGV3iKPR9ORvjUgshoyg3lNLX4ss0rHNkcfDO0DHSQvvDrtwnc+a+tseJRZUk0
GlD8xxENNsck7ucI6an9icc6bYOwZ8R8hEdbfcfZ5HKA7KzFLAuFowMrwlPBRQGH
qFGNkFnazHi8CgDRRicYd3n8hAvGGFgwYahAARaQDoQiOMLit3MKHxbF+OdPVOKe
NS9HVRCQX0jnBkLO/7192Jwd1rk2hKT6PJq5brbsQsihb+4EOU9kYWXfsxSOZChP
bdT5JJUL/qlLbXXRYqj04UHao2qK3zMsH4+I0K1N3ehCi4G+XkcqUlQWNHG8MFjw
Cq0UfogvjPdxgsfS6ci5O6SN59a7TCfHGJ5Q87d6Mtvp6XZOT4hP/felVX6pLs+Q
9hzrS7VxxdEFwj5ISQl5mg/84PRRVZOD08puovTM+mP3MAXw8m0/6mpuqXm5ze4O
un4lkX57CYcecbj/APEWUQqcEEJ2RSYutNEYKcqh5dLQu2+I4="}
----------ZYrENGBynEosq434dQEpDS
Content-Disposition: form-data; name="files"; filename="Textdatei.txt"
Content-Type: text/plain

Dies ist der Inhalt der Textdatei.
----------ZYrENGBynEosq434dQEpDS
Content-Disposition: form-data; name="files"; filename="Dokument.pdf"

```
Content-Type: application/pdf


%PDF-1..1 0 obj<</Pages 2 0 R>>endobj.2 0 obj<</Objects[3 0 R]/C

ount 1>>endobj.3 0 obj<</Folders 2 0 R>>endobj.trailer <</Root 1

 0 R>>

--------------------------ZYrENGBynEosq434dQEpDS--
```

#### 6.2.4.2 Response

```
HTTP/2 200

date: Wed, 15 May 2024 11:24:04 GMT

content-type: application/json

vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers

x-content-type-options: nosniff

cache-control: no-cache, no-store, max-age=0, must-revalidate

pragma: no-cache

expires: 0

strict-transport-security: max-age=31536000; includeSubDomains; preload

x-robots-tag: noindex, nofollow

server: Apache


{"mailboxHandle":"45d366d6-775c-4b46-8128-039866e17608","messa-
geId":1715765839805,"messageUuid":"dccb3174-4431-4102-b722-0521c07b81c8"}
```

## 6.3 Beispiele zum Einstellen von Statusmeldungen

### 6.3.1. Beispiel: Einstellen einer Statusmeldung in Java

#### 6.3.1.1 Beispiel Inhalt einer Statusmeldung als JSON

```
{
  "applicationId" : "e8bc5ebc-56cf-4ff0-a861-f989aa1ba4a9",
  "status" : "PROCESSING",
  "publicServiceName" : {
    "de" : "Anmeldung Ihres Zweitwohnsitzes"
  },
  "statusDetails" : {
    "de" : "Antrag wird bearbeitet"
  },
  "additionalInformation" : {
    "de" : "Ihr Antrag wird von einem Sachbearbeiter bearbeitet."
  },
  "reference" : "AZ123456",
  "senderName" : "Landeshauptstadt München",
  "createdDate" : "2023-11-01T11:23:21.223Z"
}
```

#### 6.3.1.2 Signieren des Inhalts der Statusmeldung

```
String signStateMessage(
        java.security.interfaces.RSAPrivateKey privateKey,
        java.lang.String stateMessage
```

```
) throws java.security.NoSuchAlgorithmException, java.security.InvalidKeyException, java.security.SignatureException {
   java.security.Signature signature = java.security.Signature.getInstance("SHA512withRSA");
   signature.initSign(privateKey);
   signature.update(stateMessage.getBytes(StandardCharsets.UTF_8));
   byte[] signed = signature.sign();
   return java.util.Base64.getEncoder().encodeToString(signed);
}
```

#### 6.3.1.3 Erstellung des Statusmeldung-Objekts und senden der Statusmeldung

```
org.springframework.http.ResponseEntity<de.akdb.egov.zbp.dto.CreateMessageResponseDTO> createStatus(
      org.springframework.web.client.RestTemplate restTemplate,
      com.fasterxml.jackson.databind.ObjectMapper objectMapper, java.net.URI createStateUrl,
      de.akdb.egov.zbp.dto.CreateStateDTO createStateDTO, java.security.interfaces.RSAPrivateKey privateKey,
      java.lang.String token) throws JsonProcessingException, java.security.NoSuchAlgorithmException,
      java.security.InvalidKeyException, java.security.SignatureException {
   java.lang.String authorToken = null;
   java.lang.String authorCertificate = null;
   java.lang.String value = objectMapper.writeValueAsString(createStateDTO);
   de.akdb.egov.zbp.dto.CreateMessageEnvelopeV6DTO signedEnvelope = new de.akdb.egov.zbp.dto.CreateMessageEnvelopeV6DTO(value,
         signStateMessage(privateKey, value), authorToken, authorCertificate);
   org.springframework.http.HttpHeaders headers = new org.springframework.http.HttpHeaders();
   headers.setBearerAuth(token);
```

```
    return restTemplate.exchange(createStateUrl, org.springframework.http.HttpMethod.PUT,
        new org.springframework.http.HttpEntity<>(signedEnvelope, headers),
        de.akdb.egov.zbp.dto.CreateMessageResponseDTO.class);
}
```

### 6.3.2 Beispiel: Einstellen einer Statusmeldung mit cURL

#### 6.3.2.1 Die Signatur berechnen

```
# Checksumme der Statusmeldung-JSON (content)
printf '{"applicationId":"93bc2fce-0242-43fa-8d1a-5a847edfc02a","status":"RECEIVED","additionalInformation":{"de":"Test 1"},"statusDetails":{"de":"Test 1"},"reference":"Test 1","publicServiceName":{"de":"Bundeswahlscheinverfahren"},"senderName":"Bundeswahlscheinverfahren","createdDate":"2024-01-14T12:23:21.223Z"}' | openssl dgst -sha512 -sign <Pfad-zum-privaten-Schlüssel> -out msg-digest.txt

# base64-Kodierung der Checksumme
base64 -w 0 msg-digest.txt

# Aufbau der Nachricht inkl. der Checksumme: Doppelte Anführungsstriche für Strings müssen mit einem \ "escaped" werden
curl -X 'POST' \
 'https://<server-host>/v6/mailbox/applications/states' \
 -H 'accept: application/json' \
 -H 'Authorization: Bearer <jwt-token>' \
 -d '{"content":"{\"applicationId\":\"93bc2fce-0242-43fa-8d1a-5a847edfc02a\",\"status\":\"RECEIVED\",\"additionalInformation\":{\"de\":\"Test 1\"},\"statusDetails\":{\"de\":\"Test 1\"},\"reference\":\"Test 1\",\"publicService-
```

Name\":{\"de\":\"Bundeswahlscheinverfahren\"},\"senderName\":\"Bundeswahlscheinverfahren\",\"createdDate\":\"2024-01-14T12:23:21.223Z\"}","sha512sum":"<output-from-base64-command>"}'

### 6.3.3. Beispiel: HTTP-Kommunikation beim Einstellen einer Statusmeldung

#### 6.3.3.1 Request

```
POST /v6/mailbox/applications/states HTTP/2
Host: zbp-pre-a-bund.akdb.de
User-Agent: curl/8.4.0
accept: */*
Authorization: Bearer eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJpY
XQiOjE3MTU3NjU3OTMsImV4cCI6MTcxNTc2NzU5Mywic2lnbmVyIjoiQmF5ZXJpc
2NoZV9FaW53b2huZXJhbmdlbGVnZW5oZWl0ZW4iLCJyb2xlcyI6WyJUSElSRF9QQ
VJUWSJdfQ.R62NZP8_Z0CkdC-Dhv5C2xHUX3o4wnrfdOAe-SrLPloKzWSER7cb91
2_RcFjtIqiU9Cbrgg-KBJwj4s0GjZ6apdKBztVDerfKcDzl4LsKFELtlLKXqHXrD
aWKjaWyCSuIN3a9reMC0QoIqQKKJ-fUYFRzyGGACLcXPtVkEpwyhxRabNc0vOv_H
q56FwyP5WQ-OAJ4vhI2vuyo0xQVee84evnrtwlhM6758VKa_VaXP6Q0Ddpz3MHSL
UMb9xrZj_ZXYSw2bVXxhwdFp02ei9etkOs9APZrbQuRKMBxekyuILoRU2A73G2V_
7efII66Ig7MnAth0TyYSeYiTp9EGcXYVoRLlodFx-AT4vKQg2P-reWua4XZuDKgJ
iY8ckGokvyQkdnDAqOKrclx4z2Z-3o3lKo8U3XLJxojBDgE7WvfhsjsJQJWJAT1S
PWw_ancYrB0p_gQFiS3f1hM05oWzUFZp63DqnNVHEMVNHdW6ZGgk-2A92ADRV8ei
IeIsgi5JIVgZ1iX6BNNj3KJvtpPlMr6n49ij8LoFJIe7NpQ4k6FhoOGqd5mM1KP9
qybrbCEtdGtYLhbycgAECLud3WR46chJtG7PLHH97GKrn9Vu6fa6bIH9tHDmkeMR
XA9zYbiMcl9qEutogaGL08ZyjObqKZv2J4Q7y6d8-_OMUMRhqyi5o
Content-Type: application/json
Content-Length: 1140
```

{"content":"{\"applicationId\":\"1ac1bffc-310d-4cf7-8c1c-772c0c9
c9082\",\"status\":\"SUBMITTED\",\"publicServiceName\":{\"de\":\
"Bayerische_Einwohnerangelegenheiten\"},\"statusDetails\":{\"de\
":\"Ihr Antrag 1ac1bffc wird bearbeitet\"},\"additionalInformati
on\":{\"de\":\"Zus..tzliche Information\"},\"senderName\":\"Baye
rische_Einwohnerangelegenheiten\",\"reference\":\"Aktenzeichen X
Y\",\"createdDate\":\"2024-05-15T09:51:36.440938599Z\"}","sha512
sum":"J2dfMNrbUt+aFvR6xIwXfUluRtuqOBtTyYOebnTT/2ZaAvkcLkbSlc3mC0
421I8YKoZIrzNh6kM38/qfi9w8CVciOrOYobUP0yl2DIUCnqsyZVuEdJ51VzlBe6
4f6Q0zEhUjtesXjD2l8ewuHgfH4868Rtc3zljfVWhVWyp38NvaWER1qRnXWlWBOw
kJJeUXqICKx33EidnRNVcmtqd7S+ohH79bgN4tfyoonnRqIvfHYGVQc4Zarfjjr0
ugxT9j0q/V5aE9mwQlg+VYcgfzAO+Q6INY+Ohh107yBiSJfgvGTfdDXlmadI+Vmy
Co2mGbutAlabS5kTpa1P3y08NtprQaQAAaypfrwbmj6ecEUI4bHUwRpBNys+JWvu
C4fXVgD8kTDyUCv7jTINwmkvqKharNGt2k40zy42GtEXj863EGPpI96yRhpxoC+s
ueu/2gic4+QVHi5J0zosUFWFGDp5wuQhjLs0OiK+wxzJ1fgnsWjDHhHf0sRtf4jR
r4gVbD5z3eBYOHWjJHcFK/xzXmKjpF5jFehsIplMQD1SynoTNsm3nobIbv+uO4KF
6kVaeRKr9ZLyY8TW6OSvOlTCNAwZV3mGUoWqmKlGRjr24p/pB62jhRtqCJyS437h
vsY/o2ktg1yYVsgkKi6iCWw0J5hK6i++ZcPYUlOt6njPzjdiU="}

#### 6.3.3.2 Response

```
HTTP/2 200
date: Wed, 15 May 2024 11:21:12 GMT
content-length: 0
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

```
x-content-type-options: nosniff
cache-control: no-cache, no-store, max-age=0, must-revalidate
pragma: no-cache
expires: 0
strict-transport-security: max-age=31536000; includeSubDomains; preload
x-robots-tag: noindex, nofollow
server: Apache
```

## 6.4 Beispiel für einen nichtfachlichen Fehlercode

```
HTTP/2 413
content-lenght: 578
content-type: text/html
date: TUE,26 Mar 2024 07:32:20 GMT
server: Apache
strict-transport-security: max-age=31536000; includeSubDomains; preload

<html>
<head><title>413 Request Entity Too Large</title></head>
<body>
<center><h1>413 Request Entity Too Large</h1></center>
<hr><center>nginx</center>
</body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

```
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
```

ozg-umsetzung.de