

# UCRI2-App Transportschicht-Nachrichten

- [Überblick](#)
- [Partielle Umsetzung](#)
- [App-Nachrichten](#)
  - [Verwendete Datentypen](#)
    - [Wahrheitswerte - type: boolean](#)
    - [Zahlen - type: number und type: integer](#)
    - [Zeichenketten - type: string](#)
    - [Objekte - type: object](#)
    - [Listen - type: array](#)
  - [App-Nachricht message\\_delivery\\_status - Zustellungsstatus für eine andere App-Nachricht](#)
    - [Eigenschaften](#)
    - [Beispiel](#)
    - [Beispiel](#)
    - [Beispiel](#)
    - [refMessageld](#)
    - [destination](#)
    - [statusCode](#)
    - [cause](#)
      - [Eigenschaften](#)
      - [cause.code](#)
      - [cause.reason](#)
      - [cause.message](#)
    - [statusMessage](#)
    - [Schema](#)

## Überblick

Diese UCRI2-App dient der Darstellung von Transportschicht-Nachrichten. Diese App MUSS von allen UCRI2-Clients ZWINGEND umgesetzt werden. UCRI2-Clients dürfen Nachrichten aus dieser App NIEMALS selbst versenden. Nachrichten aus der Transportschicht-Nachrichten-App dürfen ausschliesslich von UCRMS generiert werden. Die Nachrichten sind wie folgt:

1. Zustellungsstatusberichte (message\_delivery\_status) dienen der Übermittlung des Zustellungsstatus von anderen App-Nachrichten. Um die Entstehung von Zyklen zu vermeiden, darf ein UCRM NIEMALS eine Zustellungs-Status-Nachricht erstellen, die auf eine andere Zustellungs-Status-Nachricht verweist!
2. Teilnehmer-Verfügbarkeits-Updates (participant\_availability\_update) dienen der Übermittlung eines geänderten Verfügbarkeitsstatus zwischen gekoppelten UCRMS. Diese Nachrichten dürfen NIEMALS an UCRI2-Clients weitergegeben werden.

## Partielle Umsetzung

Ein UCRI2-Client muss nur die Zustellungsstatusberichte (message\_delivery\_status) unterstützen.

Ein UCRI2-UCRM muss alle Nachrichten unterstützen.

## App-Nachrichten

## Verwendete Datentypen

In diesem Kapitel werden die in den Nachrichten-Schemata verwendeten Datentypen beschrieben. Die beschriebenen Datentypen kommen nicht alle auch notwendigerweise in den Schemata vor.

### Wahrheitswerte - type: boolean

Mit dem Typ boolean werden JSON-Wahrheitswerte beschrieben, die möglichen Ausprägungen sind **true** und **false**.

### Zahlen - type: number und type:integer

Im JSON-Schema werden Ganzzahlen (type:integer) und Kommazahlen (type:number) separat beschrieben, wobei als Dezimalseparator ein Punkt zum Einsatz kommt. Da in JSON selbst aber nur ein gemeinsamer Zahlentyp für Ganz- und Kommazahlen existiert, können Ganzzahlen auch mit einem Dezimalseparator angegeben werden, für den Nachkommateil gilt dann aber, dass dieser 0 sein muss (z.B. sind sowohl **1.0** als auch **1** gültige Ausprägungen für eine Ganzzahl). Für beide Typen kann der gültige Zahlenbereich mit den Attributen "minimum", "exclusiveMinimum", "maximum" und "exclusiveMaximum" eingeschränkt werden).

### Zeichenketten - type: string

Zeichenketten werden über den type: string abgebildet und immer im Unicode-Format dargestellt. Für Zeichenketten werden verschiedene Untertypen genutzt, die über die Attribute "format" und "enum" (im Falle von Enumerationen) festgelegt und im Folgenden beschrieben werden:

- **Längenbeschränkte Zeichenketten:** Über die Attribute "minLength" und "maxLength" kann die Länge der Zeichenkette beschränkt werden.
- **Enumerationen:** Mit dem Attribut "enum" kann eine Liste von gültigen Ausprägungen festgelegt werden.
- **Datumsangaben:** Mit dem Attribut "format:date" werden ISO8601-konforme tagesgenaue Datumsangaben dargestellt (z.B. "2025-12-01" für den 1.Dezember 2025). Mit dem Attribut "format:date-time" werden ISO8601-konforme Zeitstempel (z.B. "2018-11-13T20:20:39") dargestellt. Als Zeitzone ist hierbei UTC festgelegt.
- **UUIDs:** Mit dem Attribut "format:uuid" werden UUIDs dargestellt, z.B. ("550e8400-e29b-41d4-a716-446655440000").
- **Weitere formateingeschränkte Zeichenketten:** Über das Attribut "pattern" kann ein regulärer Ausdruck (regex) festgelegt werden, welcher gültige Ausprägungen beschreibt. Um die Verständlichkeit der Spezifikation zu erhöhen, werden solche regulären Ausdrücke immer im "description"-Attribut einer Eigenschaft beschrieben, z.B. würde für **pattern:"^[0-9]\*\$"** die "description" als Erklärung "Der Wert muss ausschliesslich aus Ziffern bestehen" vorhanden sein.

### Objekte - type: object

JSON-Objekte werden über "type: object" beschrieben. Hierbei sind in jedem Falle nur die im "properties"-Attribut beschriebenen Objekteigenschaften gültig, weitere Eigenschaften sind nicht zulässig, dies wird über das Attribut "unevaluatedProperties": false festgelegt.

Die obligaten Eigenschaften sind als Liste im Attribut "required" hinterlegt, alle Eigenschaften, die nicht obligat sind, sind optional.

Die Anzahl an minimal vorhandenen Eigenschaften kann über das Attribut "minProperties" festgelegt werden.

### Listen - type: array

JSON-Listen (arrays) werden über "type: array" beschrieben. Hierbei wird der Typ der Listenelemente über das Attribut "items" beschrieben. Gemischt typisierte Listen sind generell unzulässig. Im Attribut "minItems" kann die minimale Anzahl von Listenelementen beschränkt werden.

## App-Nachricht message\_delivery\_status - Zustellungsstatus für eine andere App-Nachricht

Diese Nachricht wird von einem UCRM versendet, um den Zustellungsstatus einer App-Nachricht zu kommunizieren

\$id	<a href="https://github.com/PMeV-EFUL/ucri2/raw/refs/heads/main/apps/transport_layer_messages/0.1/message_delivery_status.schema.json">https://github.com/PMeV-EFUL/ucri2/raw/refs/heads/main/apps/transport_layer_messages/0.1/message_delivery_status.schema.json</a>
\$schema	<a href="https://json-schema.org/draft/2020-12/schema">https://json-schema.org/draft/2020-12/schema</a>

## Eigenschaften

Name	Typ	Obligat?
<a href="#">refMessageld</a>	String	Ja
<a href="#">destination</a>	String	Ja
<a href="#">statusCode</a>	Integer	Ja
<a href="#">cause</a>	Object	Nein
<a href="#">statusMessage</a>	String	Nein

## Beispiel

```
{
  "refMessageId": "f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454",
  "destination": "1.2.3.4.5.6.7",
  "statusCode": 504,
  "statusMessage": "Meldung konnte innerhalb von 30 Sekunden nicht zugestellt werden und wurde verworfen"
}
```

## Beispiel

```
{
  "refMessageId": "f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454",
  "destination": "1.2.3.4.5.6.7",
  "statusCode": 500,
  "statusMessage": "Das empfangende UCRM hat einen Fehler gemeldet",
  "cause": {
    "code": 468,
    "reason": "HTTP error 401: invalid signature",
    "message": "invalid signature"
  }
}
```

## Beispiel

```
{
  "refMessageId": "f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454",
  "destination": "1.2.3.4.5.6.7",
  "statusCode": 200
}
```

**refMessageld**

Titel	Referenz-UUID
Beschreibung	Systemweit eindeutige ID der referenzierten Nachricht
Typ	String
Obligat?	Ja
Format	uuid

**destination**

Titel	Empfänger-OID
Beschreibung	Empfänger-OID
Typ	String
Obligat?	Ja
Pattern	^([0-9]+\.\?)+\$

**statusCode**

Titel	Annahme- oder Ablehnungsstatus
Beschreibung	Status-Code: 200 - Zustellbestätigung; 504 - Timeout; 502 - Fehler bei Zustellung an entferntes UCRM
Typ	Integer
Obligat?	Ja
Enum	<ul style="list-style-type: none"> <li>• 200</li> <li>• 502</li> <li>• 504</li> </ul>

**cause**

Titel	Zugrundeliegender UCRI-Fehler
Beschreibung	Zugrundeliegender vom sendenden oder entfernten UCRM gemeldeter Fehler, sollte bei statusCode 500 übermittelt werden.
Typ	Object
Obligat?	Nein

**Eigenschaften**

Name	Typ	Obligat?
<a href="#">code</a>	Integer	Ja

<a href="#">reason</a>	String	Ja
<a href="#">message</a>	String	Nein

**cause.code**

<b>Titel</b>	UCRI-Fehlercode
<b>Beschreibung</b>	Anwendungsspezifische Fehler-Code
<b>Typ</b>	Integer
<b>Obligat?</b>	Ja

**cause.reason**

<b>Titel</b>	Fehlerursache
<b>Beschreibung</b>	Fehlerursache in für Menschen verständlicher Sprache
<b>Typ</b>	String
<b>Obligat?</b>	Ja

**cause.message**

<b>Titel</b>	Fehlerbeschreibung
<b>Beschreibung</b>	Detailliertere Fehlerbeschreibung eventuell mit Angabe der Korrekturmaßnahmen in für Menschen verständlicher Sprache
<b>Typ</b>	String
<b>Obligat?</b>	Nein

**statusMessage**

<b>Titel</b>	Fehlertext
<b>Beschreibung</b>	Text mit weiteren Einzelheiten und Abhilfemaßnahmen in Bezug auf den Fehler. Dies kann einem Client-Benutzer angezeigt werden.
<b>Typ</b>	String
<b>Obligat?</b>	Nein
<b>Max Length</b>	100

**Schema**

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://github.com/PMeV-
EFUL/ucri2/raw/refs/heads/main/apps/transport_layer_messages/0.1/message_delivery_status.schema.json",
```

```

    "unevaluatedProperties": false,
    "title": "App-Nachricht message_delivery_status - Zustellungsstatus für eine andere App-Nachricht",
    "description": "Diese Nachricht wird von einem UCRM versendet, um den Zustellungsstatus einer App-
    Nachricht zu kommunizieren",
    "required": [
        "refMessageId",
        "destination",
        "statusCode"
    ],
    "type": "object",
    "properties": {
        "refMessageId": {
            "type": "string",
            "format": "uuid",
            "title": "Referenz-UUID",
            "description": "Systemweit eindeutige ID der referenzierten Nachricht"
        },
        "destination": {
            "type": "string",
            "pattern": "^[0-9]+\\.?.?+$",
            "title": "Empfänger-OID",
            "description": "Empfänger-OID"
        },
        "statusCode": {
            "title": "Annahme- oder Ablehnungsstatus",
            "description": "Status-Code: 200 - Zustellbestätigung; 504 - Timeout; 502 - Fehler bei
            Zustellung an entferntes UCRM",
            "type": "integer",
            "enum": [
                200,
                502,
                504
            ]
        },
        "cause": {
            "required": [
                "code",
                "reason"
            ],
            "type": "object",
            "properties": {
                "code": {
                    "title": "UCRI-Fehlercode",
                    "type": "integer",
                    "description": "Anwendungsspezifische Fehler-Code"
                },
                "reason": {
                    "title": "Fehlerursache",
                    "type": "string",
                    "description": "Fehlerursache in für Menschen verständlicher Sprache"
                }
            },
            "message": {

```

```

        "title": "Fehlerbeschreibung",
        "type": "string",
        "description": "Detailliertere Fehlerbeschreibung eventuell mit Angabe der
Korrektivmaßnahmen in für Menschen verständlicher Sprache"
    }
},
    "title": "Zugrundeliegender UCRI-Fehler",
    "description": "Zugrundeliegender vom sendenden oder entfernten UCRM gemeldeter Fehler, sollte
bei statusCode 500 übermittelt werden."
},
    "statusMessage": {
        "title": "Fehlertext",
        "description": "Text mit weiteren Einzelheiten und Abhilfemaßnahmen in Bezug auf den Fehler.
Dies kann einem Client-Benutzer angezeigt werden.",
        "type": "string",
        "maxLength": 100
    }
},
    "examples": [
        {
            "refMessageId": "f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454",
            "destination": "1.2.3.4.5.6.7",
            "statusCode": 504,
            "statusMessage": "Meldung konnte innerhalb von 30 Sekunden nicht zugestellt werden und wurde
verworfen"
        },
        {
            "refMessageId": "f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454",
            "destination": "1.2.3.4.5.6.7",
            "statusCode": 500,
            "statusMessage": "Das empfangende UCRM hat einen Fehler gemeldet",
            "cause": {
                "code": 468,
                "reason": "HTTP error 401: invalid signature",
                "message": "invalid signature"
            }
        },
        {
            "refMessageId": "f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454",
            "destination": "1.2.3.4.5.6.7",
            "statusCode": 200
        }
    ]
}

```