



Freie
Hansestadt
Bremen



Onlinezugangsgesetz



Schnittstellenspezifikation

Teilprojekt

„Digitale Bestellung / Lieferant Cockpit“

(EfA-Umsetzungsprojekt „Vergabe“ der Freien Hansestadt Bremen)

Stand: 20.09.2023 | Version: 2.0



Autor:innen

- Alexander Ruhland,]init[AG
- Christoph Bleh,]init[AG

Vorbemerkung

Dieses Dokument dient als technische Dokumentation der Schnittstelle des Lieferantencockpits. Inhalt ist die Schnittstelle, die den Fachverfahren (Katalogsystemen) der nachnutzenden Länder zur Verfügung gestellt wird.

Bei der Schnittstelle handelt es sich um eine REST-Schnittstelle, die verschiedene APIs (Application Programming Interface) anbietet (z. B. für die Geschäftsprozesse Katalogmanagement, Beratung oder Bestellung). Welche der APIs durch den Mandanten eines Katalogsystems implementiert werden, ist abhängig von verschiedenen Faktoren (z. B. dem Funktionsumfang des Fachverfahrens oder dem Umfang der Nachnutzung durch das jeweilige Land).

Die APIs sind daher einzeln auswählbar und werden in diesem Dokument separat beschrieben. Die in den APIs enthaltenen API-Endpunkte (z. B. Beratungsanfragen empfangen) können vom anzubindenden Katalogsystem angesprochen werden. API-Endpunkte, die Daten bereitstellen, sollten in regelmäßigen Abständen angefragt werden.

Die REST-Schnittstelle ist für die Umgebungen „Staging“ und „Produktion“ verfügbar. Hierbei unterscheiden sich die Hostnamen, weshalb in den nachfolgenden Beschreibungen auf die Basis-URL der API verwiesen wird. Folgende Basis-URLs stehen zur Verfügung:

- Staging: <https://api-stage.lieferantencockpit.de>
- Produktion: <https://api.lieferantencockpit.de>

Hinweis: Die Informationen in diesem Dokument richten sich an eine technische Leserschaft. Deshalb wird das entsprechende Fachvokabular vorausgesetzt.

Die nachfolgende Abbildung stellt die im Dokument beschriebene REST-Schnittstelle mit den enthaltenen APIs und den jeweils angebotenen API-Endpunkten dar.

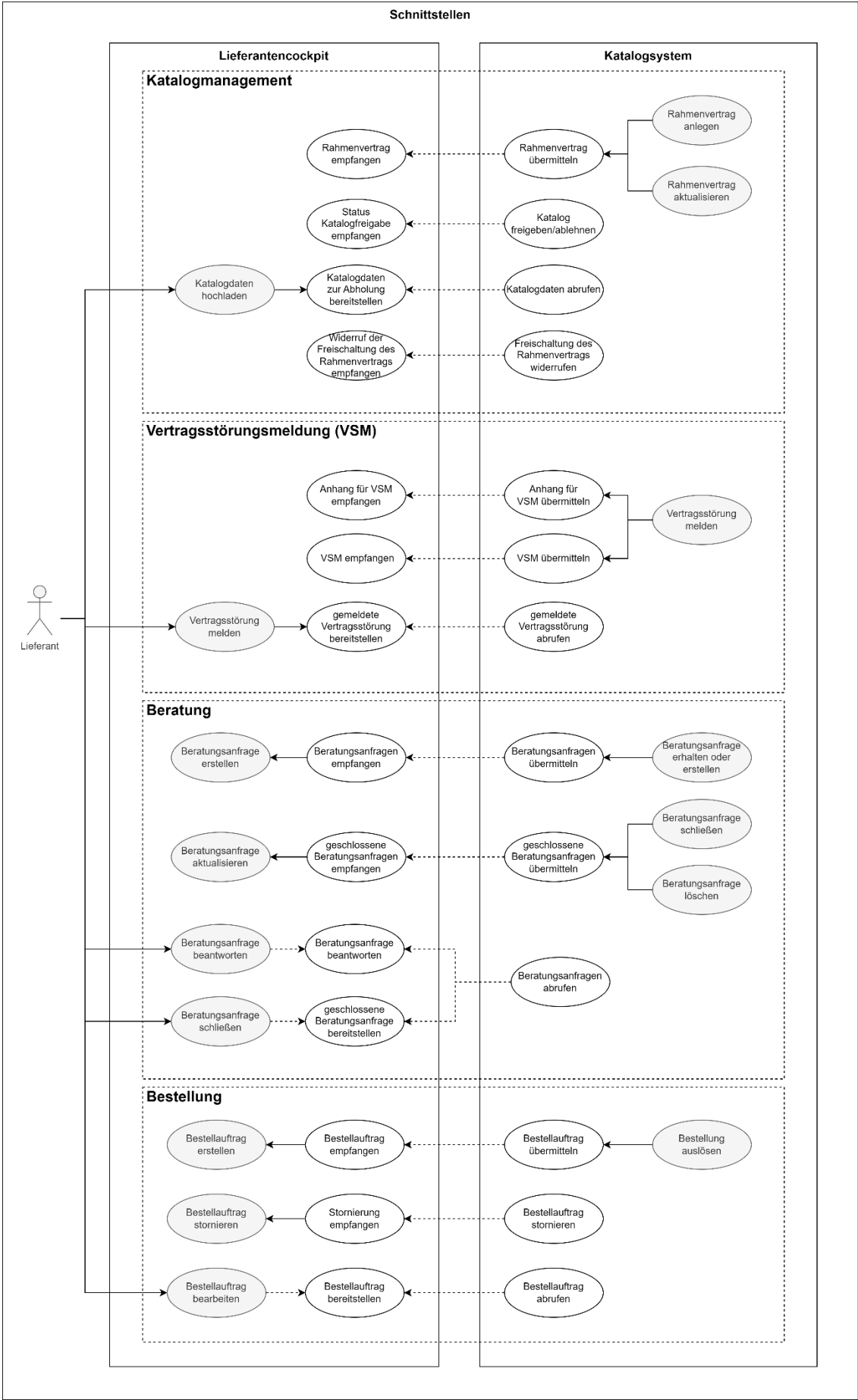


Abbildung 1: Schnittstelle „Katalogsystem“

Änderungsverzeichnis

Datum	Beschreibung der Änderung	Mitwirkende, Unternehmen
26.07.2022	<p>Kapitel 10</p> <ul style="list-style-type: none"> Form-Data Teil "Bestellung" in "Metadaten" umbenannt Form-Data Teil "XBestellung" in "Bestellung" umbenannt Unterobjekt "Bestellung" aus JSON entfernt Unterobjekt "Konfiguration" hinzugefügt Beispiel für ein strukturiertes JSON als Dateiformat hinzugefügt <p>Kapitel 12, 13 und 14: neu</p>	Alexander Ruhland,]init[AG Pascal Mai ,]init[AG
29.07.2022	<p>Kapitel 10</p> <ul style="list-style-type: none"> Beispiel: „JSON Endpunkt „Bestellungen empfangen“ – Teil Bestellung“ aktualisiert Attribut-Definitionen aktualisiert 	Alexander Ruhland,]init[AG
04.08.2022	<p>Kapitel 15: Beantragung von Zugangsdaten hinzugefügt</p> <p>Kapitel 16: Sichere Kommunikation aktualisiert</p> <p>Kapitel 2-4:</p> <ul style="list-style-type: none"> Routen korrigiert (cs als Teil des Pfades ergänzt) API-Beschreibung angepasst (Authorization-Header) 	Alexander Ruhland,]init[AG
21.11.2022	<p>Kapitel 2-14:</p> <ul style="list-style-type: none"> API-Beschreibung angepasst Attribut-Definitionen angepasst Beispiele angepasst <p>Kapitel 11: neu</p> <p>Kapitel 17:</p> <ul style="list-style-type: none"> Abbildungen der Beispiele aktualisiert Übergreifende Abbildungen aktualisiert 	Alexander Ruhland,]init[AG Christoph Bleh,]init[AG
10.05.2023	<p>Kapitel 2-15:</p> <ul style="list-style-type: none"> API-Beschreibung angepasst Umstellung auf LieferantencockpitId eingefügt Beispiele angepasst <p>Kapitel 4:</p> <ul style="list-style-type: none"> Neu 	Alexander Ruhland,]init[AG
17.08.2023	<p>Kapitel 3-18:</p> <ul style="list-style-type: none"> „Mein Unternehmenskonto“ einheitlich benannt Kleinere Korrekturen Abbildungen aktualisiert <p>Kapitel 7: neu</p>	Alexander Ruhland,]init[AG

31.08.2023	Kapitel 3-18: <ul style="list-style-type: none"> • kleinere Textanpassungen 	Alexander Ruhland,]init[AG
04.-05.09.2023	Formale Prüfung und Qualitätssicherung	Daniela Riegler, Nortal AG; David Groten,]init[AG; Marit Jensen,]init[AG
15.09.2023	Fehlerkorrektur und Formatanpassungen	Alexander Ruhland,]init[AG
20.09.2023	Freigabe erteilt	Torsten Masuhr, Senator für Finanzen Bremen

Inhaltsverzeichnis

Autor:innen	2
Vorbemerkung.....	3
Änderungsverzeichnis	5
Abkürzungsverzeichnis	11
Abbildungsverzeichnis.....	12
Tabellenverzeichnis	13
Codeblöcke	15
1. Beantragung von Zugangsdaten	16
2. Sichere Kommunikation	17
3. Rahmenvertragsdaten empfangen.....	19
3.1. Sichere Kommunikation	19
3.2. API-Beschreibung	19
3.2.1. POST Body	19
3.3. Attribut-Definitionen.....	20
3.4. Response	22
4. Rahmenvertragsdaten aktualisieren	23
4.1. Sichere Kommunikation	23
4.2. API-Beschreibung	23
4.2.1. POST Body	23
4.3. Attribut-Definitionen.....	24
4.4. Response	26
5. Katalogbereitstellung	27
5.1. Sichere Kommunikation	27
5.2. API-Beschreibung	27
5.2.1. Katalogbereitstellung prüfen.....	27
5.2.1.1. Response Codes.....	27
5.2.1.2. Response Payload.....	28
5.2.1.3. Attribut-Definitionen.....	29
5.2.2. Katalogdatei abholen	30
5.2.2.1. Response Codes.....	30
5.2.2.2. Response Payload.....	31
5.2.3. Mediadatei abholen	31
5.2.3.1. Response Codes.....	31

5.2.3.2.	Response Payload.....	32
5.2.4.	Katalogverarbeitung – Statuscodes.....	32
6.	Statusanpassungen empfangen	33
6.1.	Sichere Kommunikation	33
6.2.	API-Beschreibung	33
6.2.1.	POST Body	33
6.3.	Attribut-Definitionen.....	34
6.4.	Response Codes.....	34
6.5.	Mögliche Statusangaben	35
7.	Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen	37
7.1.	Sichere Kommunikation	37
7.2.	API-Beschreibung	37
7.2.1.	POST Body	37
7.3.	Attribut-Definitionen.....	37
7.4.	Response	38
8.	Beratungsanfragen empfangen	40
8.1.	Sichere Kommunikation	40
8.2.	API-Beschreibung	40
8.2.1.	POST Body	40
8.3.	Attribut-Definitionen.....	41
8.4.	Response	43
9.	Beratungsanfragen beantwortet	44
9.1.	Sichere Kommunikation	44
9.2.	API-Beschreibung	44
9.3.	Response Codes.....	44
9.4.	Response Payload.....	45
9.5.	Attribut-Definitionen.....	46
10.	Geschlossene Beratungsanfragen empfangen	48
10.1.	Sichere Kommunikation	48
10.2.	API-Beschreibung	48
10.2.1.	POST Body	48
10.3.	Attribut-Definitionen.....	49
10.4.	Response	49
11.	Geschlossene Beratungsanfrage bereitstellen	51

11.1.	Sichere Kommunikation	51
11.2.	API-Beschreibung	51
11.3.	Response Codes.....	51
11.4.	Response Payload.....	52
11.5.	Attribut-Definitionen.....	53
12.	Bestellung bereitstellen.....	54
12.1.	Sichere Kommunikation	54
12.2.	API-Beschreibung	54
12.3.	Response Codes.....	54
12.4.	Response Payload.....	56
12.5.	Attribut-Definitionen.....	56
12.5.1.	Anhang zum Bestellauftrag abholen	57
12.5.2.	Response Payload.....	57
12.6.	Mögliche Statusangaben.....	57
13.	Bestellung empfangen.....	59
13.1.	Sichere Kommunikation	59
13.2.	API-Beschreibung	59
13.2.1.	Request (Teil Metadaten).....	60
13.2.1.1.	Attribut-Definitionen.....	60
13.2.2.	Request (Teil Bestellung).....	61
13.2.2.1.	Attribut-Definitionen.....	64
13.2.3.	Response Codes.....	68
13.3.	Response Payload.....	69
14.	Bestellung stornieren	70
14.1.	Sichere Kommunikation	70
14.2.	API-Beschreibung	70
14.2.1.	Request Body.....	70
14.3.	Attribut-Definitionen.....	71
14.4.	Response	71
15.	Vertragsstörungsmeldung (VSM) empfangen.....	73
15.1.	Sichere Kommunikation	73
15.2.	API-Beschreibung	73
15.2.1.	Request Body.....	73
15.3.	Attribut-Definitionen.....	74

15.4. Response	75
16. Vertragsstörungsmeldung (VSM) Anhang empfangen.....	77
16.1. Sichere Kommunikation	77
16.2. API-Beschreibung	77
16.2.1. Request Body.....	77
16.3. Response Codes.....	77
16.3.1. Response Payload.....	78
17. Vertragsstörungsmeldung bereitstellen.....	79
17.1. Sichere Kommunikation	79
17.2. API-Beschreibung	79
17.3. Response Codes.....	79
17.4. Response Payload.....	80
17.5. Attribut-Definitionen.....	81
18. Anlagen	83
18.1. Übergreifende Abbildungen	83
18.2. Abbildungen Katalogmanagement.....	84
18.3. Abbildungen Beratung.....	89
18.4. Abbildungen Bestellung	93

Abkürzungsverzeichnis

Abkürzung/Akronym	Bedeutung
API	Application Programming Interface
DNS	Domain Name System
GUID	Globally Unique Identifier
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
PROD	Produktivumgebung
REST	Representational State Transfer
STAGE	Stage-Umgebung
URL	Uniform Resource Locator
VSM	Vertragsstörungsmeldung

Abbildungsverzeichnis

Abbildung 1: Schnittstelle „Katalogsystem“	4
Abbildung 2: Beispiel für die Konfiguration zur Generierung von Access Tokens	18
Abbildung 3: Workflow des Katalogmanagements	36
Abbildung 4: High Level Architektur des Lieferantencockpits (vergrößerte Darstellung)	83
Abbildung 5: Workflow des Katalogmanagements (vergrößerte Darstellung)	84
Abbildung 6: Insomnia-Beispiel der API „Rahmenvertragsdaten empfangen“	85
Abbildung 7: Insomnia-Beispiel der API „Rahmenvertragsdaten aktualisieren“	86
Abbildung 8: Insomnia-Beispiel der API „Katalogbereitstellung“	87
Abbildung 9: Insomnia-Beispiel der API „Statusanpassungen empfangen“	88
Abbildung 10: Insomnia-Beispiel der API „Beratungsanfragen empfangen“	89
Abbildung 11: Insomnia-Beispiel der API „Beratungsanfragen beantwortet“	90
Abbildung 12: Insomnia-Beispiel der API „Geschlossene Beratungsanfragen empfangen“	91
Abbildung 13: Insomnia-Beispiel der API „Geschlossene Beratungsanfragen bereitstellen“	92
Abbildung 14: Insomnia-Beispiel der API „Bestellung bereitstellen“	93
Abbildung 15: Insomnia-Beispiel der API „Bestellung empfangen“	94
Abbildung 16: Insomnia-Beispiel der API „Bestellung stornieren“	95

Tabellenverzeichnis

Tabelle 1: Attribut-Definitionen der API „Rahmenvertragsdaten empfangen“	22
Tabelle 2: Mögliche Response Codes der API „Rahmenvertragsdaten empfangen“	22
Tabelle 3: Attribut-Definitionen der API „Rahmenvertragsdaten aktualisieren“	26
Tabelle 4: Mögliche Response Codes der API „Rahmenvertragsdaten aktualisieren“	26
Tabelle 5: Mögliche Response Codes im Rahmen der Prüfung der Katalogdaten	28
Tabelle 6: Attribut-Definitionen der API „Katalogbereitstellung“	30
Tabelle 7: Mögliche Response Codes im Rahmen der Abholung von Katalogdateien.....	31
Tabelle 8: Dateiausgabe der Katalogdatei.....	31
Tabelle 9: Mögliche Response Codes im Rahmen der Abholung von Mediadateien	32
Tabelle 10: Dateiausgabe der Mediadatei	32
Tabelle 11: Statuscodes der Katalogverarbeitung	32
Tabelle 12: Attribut-Definitionen der API „Statusanpassung empfangen“	34
Tabelle 13: Mögliche Response Codes der API „Statusanpassungen empfangen“	35
Tabelle 14: Mögliche Statusangaben	36
Tabelle 15: Attribut-Definitionen der API „Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen“	38
Tabelle 16: Mögliche Response Codes der API „Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen“	39
Tabelle 17: Attribut-Definitionen der API „Beratungsanfragen empfangen“	42
Tabelle 18: Mögliche Response Codes der API „Beratungsanfragen empfangen“	43
Tabelle 17: Optionale Query-Parameter der API „Beratungsanfragen beantworten“	44
Tabelle 20: Mögliche Response Codes der API „Beratungsanfragen beantworten“	45
Tabelle 21: Attribut-Definitionen der API „Beratungsanfragen beantworten“	47
Tabelle 22: Attribut-Definitionen der API „Geschlossene Beratungsanfragen empfangen“	49
Tabelle 23: Mögliche Response Codes der API „Geschlossene Beratungsanfragen empfangen“	50
Tabelle 24: Optionale Query-Parameter der API „Geschlossene Beratungsanfragen bereitstellen“ ...	51
Tabelle 25: Mögliche Response Codes der API „Geschlossene Beratungsanfragen bereitstellen“	52
Tabelle 26: Attribut-Definitionen der API „Geschlossene Beratungsanfragen bereitstellen“	53
Tabelle 27: Optionale Query-Parameter der API „Bestellung bereitstellen“	54
Tabelle 28: Mögliche Response Codes der API „Bestellung bereitstellen“	55
Tabelle 29: Attribut-Definitionen der API „Bestellung bereitstellen“	57
Tabelle 30: Mögliche Response Codes der API „Bestellung bereitstellen“ - Bestellung abholen.....	57

Tabelle 31: Mögliche Statusmeldungen der API „Bestellung bereitstellen“	58
Tabelle 32: Attribut-Definitionen der API „Bestellung empfangen“ - Teil Metadaten	61
Tabelle 33: Attribut-Definitionen der API „Bestellung empfangen“ - Teil Bestellung	68
Tabelle 34: Mögliche Response Codes der API „Bestellung empfangen“	69
Tabelle 35: Attribut-Definitionen der API „Bestellung stornieren“	71
Tabelle 36: Mögliche Response Codes der API „Bestellung stornieren“	72
Tabelle 37: Attribut-Definitionen der API „VSM empfangen“	75
Tabelle 38: Mögliche Response Codes der API „VSM empfangen“	76
Tabelle 39: Mögliche Response Codes der API „VSM Anhang empfangen“	78
Tabelle 40: Optionale Query-Parameter der API „VSM bereitstellen“	79
Tabelle 41: Mögliche Response Codes der API „Vertragsstörungsmeldung bereitstellen“	80
Tabelle 42: Attribut-Definitionen der API „VSM bereitstellen“	82

Codeblöcke

Codeblock 1: Beispiel - JSON Endpunkt „Rahmenvertragsdaten empfangen“	20
Codeblock 2: Beispiel - JSON Endpunkt „Rahmenvertragsdaten aktualisieren“	24
Codeblock 3: Response - JSON Endpunkt „Katalogbereitstellungen“	29
Codeblock 4: Beispiel - JSON Endpunkt „Statusanpassungen empfangen“	33
Codeblock 5: Beispiel - JSON Endpunkt „Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen“	37
Codeblock 6: Beispiel - JSON Endpunkt „Beratungsanfragen empfangen“	41
Codeblock 7: Response - JSON Endpunkt „Beantwortete Beratungsanfragen bereitstellen“	46
Codeblock 8: Beispiel - JSON Endpunkt „Geschlossene Beratungsanfragen empfangen“	48
Codeblock 9: Response - JSON Endpunkt „Geschlossene Beratungsanfragen bereitstellen“	52
Codeblock 10: Response - JSON Endpunkt „Bestellung bereitstellen“	56
Codeblock 11: Beispiel - JSON Endpunkt „Bestellungen empfangen“ - Teil Metadaten	60
Codeblock 12: Beispiel - JSON Endpunkt „Bestellungen empfangen“ - Teil Bestellung.....	63
Codeblock 13: Beispiel - Response Payload „Bestellung empfangen“	69
Codeblock 14: Beispiel - JSON Endpunkt „Bestellung stornieren“	70
Codeblock 15: Beispiel - JSON Endpunkt „VSM empfangen“	73
Codeblock 16: Beispiel - JSON Endpunkt „VSM Anhang empfangen“	77
Codeblock 17: Beispiel - JSON Response der API „VSM Anhang empfangen“	78
Codeblock 18: Response - JSON Endpunkt „VSM bereitstellen“	81

1. Beantragung von Zugangsdaten

Zur Anbindung eines Katalogsystems an das Lieferantencockpit werden vom Fachverfahrenshersteller die nachfolgenden Informationen benötigt:

- Name des Katalogsystems
- Bundesland (in dem der Mandant des Katalogsystems zum Einsatz kommt)
- Kontaktdaten der technischen Ansprechperson
 - Vor- und Nachname
 - E-Mail-Adresse
 - Telefonnummer

Die Betreiber des Lieferantencockpits legen anschließend die Zugangsdaten für das Katalogsystem an und übermitteln diese an die angegebene Ansprechperson. Der Austausch der Zugangsdaten erfolgt per E-Mail mittels passwortgeschützter Zip-Datei. Das Passwort dieser Datei wird über einen anderen Kanal (z. B. Telefon) ausgetauscht.

Die Nutzung der Schnittstelle des Lieferantencockpits setzt voraus:

Das jeweilige Land (1.) nutzt das Lieferantencockpit nach und (2.) hat die Anbindung des Fachverfahrens beauftragt. Nur wenn diese Voraussetzungen erfüllt sind, wird die Nutzung der Schnittstelle im Rahmen eines Implementierungsprojekts abgestimmt und ermöglicht.

Je Mandant eines Katalogsystems werden separate Zugangsdaten vergeben. Der Grund dafür besteht darin, dass so je Mandant der Umfang der Nachnutzung konfiguriert werden kann (bezogen auf ein Land).

2. Sichere Kommunikation

Der Schutz vor unbefugtem Zugriff auf die Schnittstellen erfolgt mittels OAuth2.0 Client Credentials. Verbindende Katalogsysteme müssen sich sowohl über ihre Client-ID als auch ihren Client-Secret authentifizieren lassen. Als Rückgabe erhalten sie ein Access Token, das zur Autorisierung verwendet werden muss. Dieses Token ist bei jedem Aufruf der einzelnen Schnittstellen im Header mit dem Authorization-Prefix "Baerer" anzugeben.

Ein Access Token wird über die Access Token URL abgerufen. Folgende Konfiguration ist dazu notwendig:

- Access Token URL:
 - STAGE: <https://auth-stage.lieferantencockpit.de/connect/token>
 - PROD: <https://auth.lieferantencockpit.de/connect/token>
- Grant Type: Client Credentials
- Client ID: [client_id]
- Client Secret: [client_secret]

Die generierten Access Token sind eine Stunde gültig. Danach muss über den gleichen Endpunkt ein neues Token angefordert werden.

Beispiel

Die Generierung von Access Tokens per Insomnia (als Beispiel) erfolgt durch folgende Einstellungen:

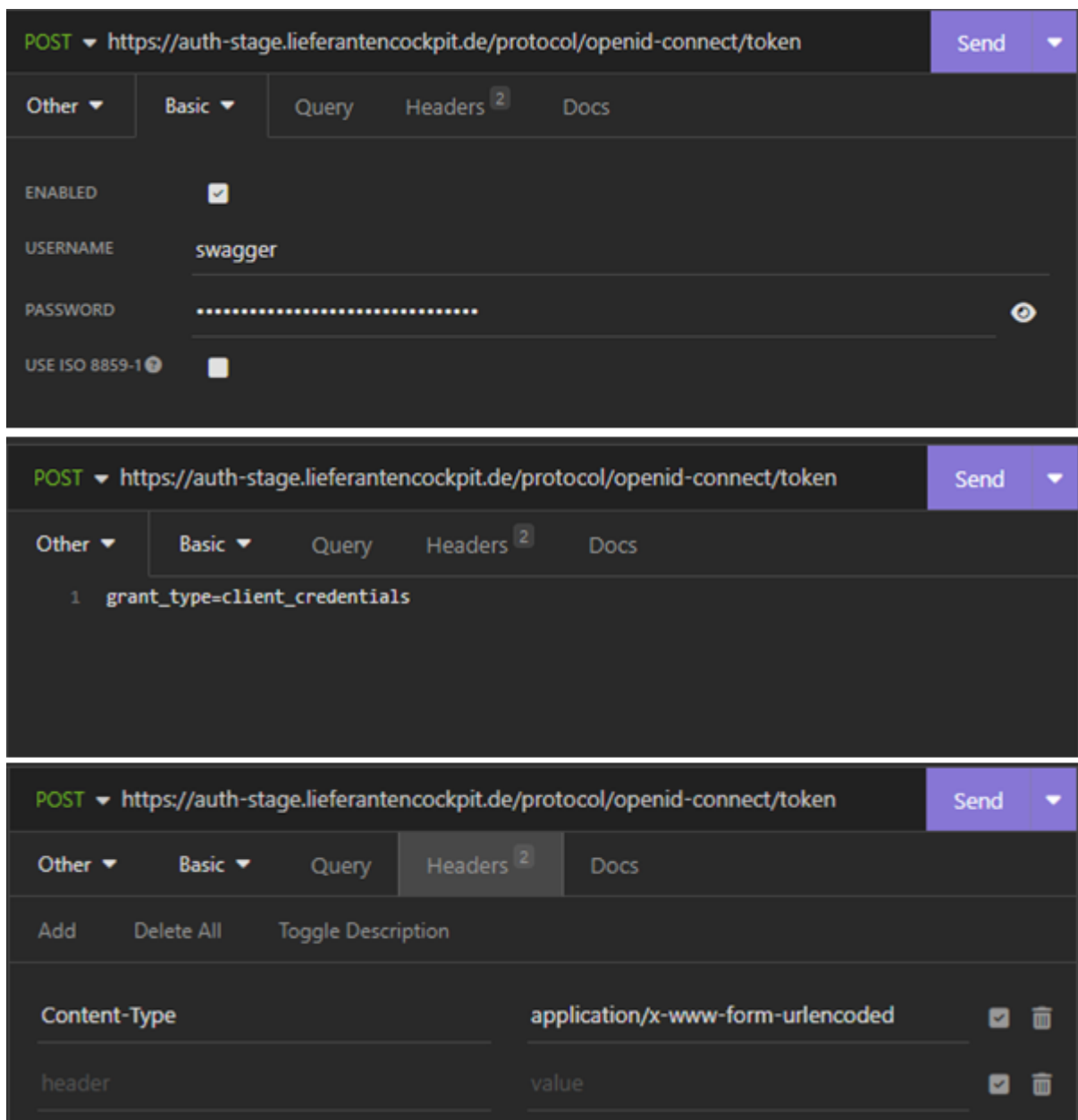


Abbildung 2: Beispiel für die Konfiguration zur Generierung von Access Tokens

3. Rahmenvertragsdaten empfangen

Über diese API werden vom Katalogsystem entsprechende Rahmenvertragsdaten an das Lieferantencockpit gesendet. Es obliegt dem Fachverfahrenshersteller festzulegen, zu welchen Anlässen und Zeitpunkten die API aufgerufen wird. Die Daten werden in einer vom Lieferantencockpit betriebenen Datenbank gespeichert und stehen somit beim Login („Mein Unternehmenskonto“) für die Abfrage anhand der Lieferantencockpit-ID bereit.

Die Aktualisierung von bereits vorhandenen Rahmenverträgen kann mithilfe der entsprechenden Schnittstelle durchgeführt werden.

3.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

3.2. API-Beschreibung

Das Entgegennehmen von Rahmenvertragsdaten erfolgt über diesen Endpunkt:

HTTP POST an: API Basis-URL + /v2/rahmenvertrag/empfangen

Authorization Header: Bearer + Access Token

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 6: Insomnia-Beispiel der API „Rahmenvertragsdaten empfangen“).

3.2.1. POST Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "katalogSystem": {
    "rahmenvertrag": {
      "rahmenvertragId": "ad1a02c7-df04-4fa4-b0e1-209485657881",
      "vertragsnummer": "RV-001-001",
      "vertragsbezeichnung": "Test Rahmenvertrag",
      "vertragslaufzeitStart": "2021-01-01T00:00:00.000Z",
      "vertragslaufzeitEnde": "2021-12-31T00:00:00.000Z",
      "ansprechpersonName": "Max Mustermann",
      "ansprechpersonEmail": "max@mustermann.de",
      "ansprechpersonTelefon": "+49123456789"
    },
    "lieferanten": [
      {
        "lieferantencockpitId": "56798a4e-0178-4270-b137-c4ece52bc810",
        "lose": [
          {
            "losId": "0978c1f7-f89b-4e8f-931d-ed43d1500cdb"
          },
          {
            "losId": "f77cf830-167b-4cf3-ab64-68c13492527d"
          }
        ]
      },
      {

```

```

    "lieferantcockpitId": "21ff8f59-05cc-447d-849f-c88b1d53cbb4",
    "lose": [
      {
        "losId": "ac5bd952-8494-4b6d-9564-580d1eccac5e"
      }
    ]
  },
  "rahmenvertragspartner": {
    "rahmenvertragspartnerId": "RVP-001",
    "name": "Test",
    "strasse": "Teststr.",
    "hausnummer": "123",
    "plz": "12345",
    "stadt": "teststadt"
  }
}
}
}

```

Codeblock 1: Beispiel - JSON Endpunkt „Rahmenvertragsdaten empfangen“

3.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Rahmenvertrag			
rahmenvertragsId	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	GUID	ja
Vertragsnummer	Rahmenvertragsnummer des verlinkten Rahmenvertrags	String	nein
vertragsbezeichnung	Bezeichner des Rahmenvertrags	String	nein
vertragslaufzeitStart	Beginn der Vertragslaufzeit	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)	nein
vertragslaufzeitEnde	Ende der Vertragslaufzeit	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)	nein

ansprechpersonName	Name der Ansprechperson	String	nein
ansprechpersonEmail	E-Mail der Ansprechperson	String	nein
ansprechpersonTelefon	Telefonnummer der Ansprechperson	String	nein
Lieferanten			
Lieferant			
lieferantencockpitId	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	ja
Lose			
Los			
losId	eindeutiges Identifikationsmerkmal des Loses Diese GUID ist notwendig, um eine eindeutige Beziehung zwischen Rahmenvertrag, Lieferanten und Los herzustellen.	GUID	ja
Rahmenvertragspartner			
rahmenvertragspartnerId	Die RahmenvertragspartnerId ist eine eindeutige ID für die jeweiligen Organisationen und Behörden. Falls im Katalogsystem bereits ein Wert hierfür existiert, kann dieser von der Schnittstelle übernommen werden.	String	ja
name	Name des Rahmenvertragspartners	String	nein
strasse	Straßenname des Rahmenvertragspartners	String	nein
hausnummer	Hausnummer des Rahmenvertragspartners	String	nein
plz	Postleitzahl des Rahmenvertragspartners	String	nein

stadt	Name der Stadt des Rahmenvertragspartners	String	nein
-------	-------------------------------------------	--------	------

Tabelle 1: Attribut-Definitionen der API „Rahmenvertragsdaten empfangen“

3.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes lauten:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiuupload 4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	<p>Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt.</p> <p>Das Format der übermittelten Daten ist nicht mit dem Datenformat des Servers kompatibel.</p>
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 2: Mögliche Response Codes der API „Rahmenvertragsdaten empfangen“

4. Rahmenvertragsdaten aktualisieren

Über diese API werden vom Katalogsystem entsprechende Rahmenvertragsdaten an das Lieferantencockpit zur Aktualisierung gesendet. Es obliegt dem Aufrufer festzulegen, zu welchen Anlässen und Zeitpunkten die API aufgerufen wird. Diese Daten überschreiben den vorhandenen Datensatz und stehen somit beim Login („Mein Unternehmenskonto“) für die Abfrage anhand der Lieferantencockpit-ID bereit.

Die Aktualisierung von bereits vorhandenen Rahmenverträgen wird durch Abgleich des Parameters für den Identifikator des Rahmenvertrags mit dem Namen „rahmenvertragId“ ermöglicht.

4.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

4.2. API-Beschreibung

Die Aktualisierung von Rahmenvertragsdaten erfolgt über diesen Endpunkt:

HTTP PUT an: API Basis-URL + /v2/rahmenvertrag/aktualisieren

Authorization Header: Bearer + Access Token

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 7: Insomnia-Beispiel der API „Rahmenvertragsdaten aktualisieren“).

4.2.1. POST Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "katalogSystem": {
    "rahmenvertrag": {
      "rahmenvertragId": "ad1a02c7-df04-4fa4-b0e1-209485657881",
      "vertragsnummer": "RV-001-001",
      "vertragsbezeichnung": "Test Rahmenvertrag",
      "vertragslaufzeitStart": "2021-01-01T00:00:00.000Z",
      "vertragslaufzeitEnde": "2021-12-31T00:00:00.000Z",
      "ansprechpersonName": "Max Mustermann",
      "ansprechpersonEmail": "max@mustermann.de",
      "ansprechpersonTelefon": "+49123456789"
    },
    "lieferanten": [
      {
        "lieferantencockpitId": "56798a4e-0178-4270-b137-c4ece52bc810",
        "lose": [
          {
            "losId": "0978c1f7-f89b-4e8f-931d-ed43d1500cdb"
          },
          {
            "losId": "f77cf830-167b-4cf3-ab64-68c13492527d"
          }
        ]
      },
      {

```

```

    "lieferantencockpitId": "21ff8f59-05cc-447d-849f-c88b1d53cbb4",
    "lose": [
      {
        "losId": "ac5bd952-8494-4b6d-9564-580d1eccac5e"
      }
    ]
  },
  "rahmenvertragspartner": {
    "rahmenvertragspartnerId": "RVP-001",
    "name": "Test",
    "strasse": "Teststr.",
    "hausnummer": "123",
    "plz": "12345",
    "stadt": "teststadt"
  }
}
}
}

```

Codeblock 2: Beispiel - JSON Endpunkt „Rahmenvertragsdaten aktualisieren“

4.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Rahmenvertrag			
rahmenvertragsId	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	GUID	ja
vertragsnummer	Rahmenvertragsnummer des verlinkten Rahmenvertrags	String	nein
vertragsbezeichnung	Bezeichner des Rahmenvertrags	String	nein
vertragslaufzeitStart	Beginn der Vertragslaufzeit	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)	nein
vertragslaufzeitEnde	Ende der Vertragslaufzeit	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)	nein

ansprechpersonName	Name der Ansprechperson	String	nein
ansprechpersonEmail	E-Mail der Ansprechperson	String	nein
ansprechpersonTelefon	Telefonnummer der Ansprechperson	String	nein
Lieferanten			
Lieferant			
lieferantencockpitId	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	ja
Lose			
Los			
losId	eindeutiges Identifikationsmerkmal des Loses Diese GUID ist notwendig, um eine eindeutige Beziehung zwischen Rahmenvertrag, Lieferanten und Los herzustellen.	GUID	ja
Rahmenvertragspartner			
rahmenvertragspartnerId	Die RahmenvertragspartnerId ist eine eindeutige ID für die jeweiligen Organisationen und Behörden. Falls im Katalogsystem bereits ein Wert hierfür existiert, kann dieser von der Schnittstelle übernommen werden.	String	ja
name	Name des Rahmenvertragspartners	String	nein
strasse	Straßenname des Rahmenvertragspartners	String	nein
hausnummer	Hausnummer des Rahmenvertragspartners	String	nein
plz	Postleitzahl des Rahmenvertragspartners	String	nein

stadt	Name der Stadt des Rahmenvertragspartners	String	nein
-------	-------------------------------------------	--------	------

Tabelle 3: Attribut-Definitionen der API „Rahmenvertragsdaten aktualisieren“

4.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes lauten:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiuupload 4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	<p>Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt.</p> <p>Das Format der übermittelten Daten ist nicht mit dem Datenformat des Servers kompatibel.</p>
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 4: Mögliche Response Codes der API „Rahmenvertragsdaten aktualisieren“

5. Katalogbereitstellung

Über diese API können Katalogsysteme bereits bereitgestellte Katalogdateien abrufen. Eine JSON-Response enthält die bereitgestellten Katalogdateien und deren zugehörige Katalogfreigaben. Diese Katalogfreigaben werden für die Statusaktualisierung per API „Statusanpassungen empfangen“ benötigt, damit Statusanpassungen an das Lieferantencockpit gemeldet werden können.

Es werden gesonderte Endpunkte für Katalogdateien und Mediadateien bereitgestellt, an die eine GUID der Katalogfreigabe übermittelt werden muss. Somit werden die zugehörigen Dateien herausgegeben.

Nach dem Erhalt der JSON-Response können die jeweiligen Katalogdateien vereinzelt über die mitgegebenen URLs heruntergeladen werden.

5.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

5.2. API-Beschreibung

Diese API dient zur Abholung von Katalogdaten, für die eine Katalog- und/oder Mediadatei von den Lieferanten hochgeladen wurde, aber deren Abholung noch nicht vom Katalogsystem durch Setzen des Status 5 oder 6 (siehe Tabelle 11: Statuscodes der Katalogverarbeitung) bestätigt wurde. Sie besteht aus drei Teilen: Katalogbereitstellung prüfen, Katalogdatei abholen, Mediadatei abholen.

5.2.1. Katalogbereitstellung prüfen

Die Prüfung auf aktuell verfügbare Katalogdaten erfolgt über diesen Endpunkt:

HTTP GET an: API Basis-URL + /v2/rahmenvertrag/freigegebenekataloge

Authorization Header: Bearer + Access Token

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 8: Insomnia-Beispiel der API „Katalogbereitstellung“).

5.2.1.1. Response Codes

Mögliche Response Codes lauten:

Response Code	Mögliche Ursachen
200 OK	Die angeforderten Daten konnten erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies

	<p>3. zu große Dateien bei Dateiuupload</p> <p>4. asynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)</p>
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
406 Method Not Acceptable	Nach Client-Request ist der Server nicht in der Lage, die vom Client im Accept-Header geforderten Formate zu senden.
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 5: Mögliche Response Codes im Rahmen der Prüfung der Katalogdaten

5.2.1.2. Response Payload

Der ausgegebene Datensatz entspricht folgender Struktur (Beispiel):

```

{
  "katalogsystem": {
    "katalogfreigaben": [
      {
        "freigabeId": "df69b42b-b6b5-4dc8-9afa-ebd6f43b4586",
        "losId": "0978c1f7-f89b-4e8f-931d-ed43d1500cdb",
        "katalogFile": "https://api-stage.lieferantcockpit.de/cs/rahmenvertrag/kataloge/df69b42b-b6b5-4dc8-9afa-ebd6f43b4586",
        "medienFile": "https://api-stage.lieferantcockpit.de/cs/rahmenvertrag/medien/df69b42b-b6b5-4dc8-9afa-ebd6f43b4586",
        "verarbeitungsstatus": 0,
        "rahmenvertrag": {
          "rahmenvertragId": "ad1a02c7-df04-4fa4-b0e1-209485657881",
          "vertragsnummer": "RV-001-001"
        },
        "lieferant": {
          "lieferantcockpitId": "463f93a9-f293-449c-9c0c-ca511f1aa17a"
        }
      },
      {
        "freigabeId": "e112b7f2-41ee-427e-9390-763110b922b2",
        "losId": "ac5bd952-8494-4b6d-9564-580d1eccac5e",
        "katalogFile": "https://api-stage.lieferantcockpit.de/cs/rahmenvertrag/kataloge/e112b7f2-41ee-427e-9390-763110b922b2",

```

```

    "medienFile": "https://api-stage.lieferantencockpit.de/cs/rahmenver-
trag/medien/e112b7f2-41ee-427e-9390-763110b922b2",
    "verarbeitungsstatus": 3,
    "rahmenvertrag": {
      "rahmenvertragId": "91e91d9c-b374-420f-9374-fa94ba60b5b0",
      "vertragsnummer": "RV-002-002"
    },
    "lieferant": {
      "lieferantencockpitId": "21ff8f59-05cc-447d-849f-c88b1d53cbb4"
    }
  }
]
}

```

Codeblock 3: Response - JSON Endpunkt „Katalogbereitstellungen“

5.2.1.3. Attribut-Definitionen

Die im Datensatz ausgegebenen Werte beziehen sich auf die nachfolgenden Datenfelder. Dieser Endpunkt gibt lediglich Daten zurück und hat demnach keine Pflichtfelder.

Attribut	Beschreibung	Typ	Pflichtfeld
Katalogfreigabe			
freigabeld	eindeutiges Identifikationsmerkmal der Katalogfreigabe Diese GUID ist notwendig, um eine eindeutige Beziehung zwischen Katalogdateien, Mediendateien und Katalogfreigabe herzustellen.	GUID	Nein
losId	eindeutiges Identifikationsmerkmal des Loses Diese GUID ist notwendig, um eine eindeutige Beziehung zwischen Rahmenvertrag, Lieferanten und Los herzustellen.	GUID	Nein
katalogFile	URL zur Abholung der Katalogdatei Sie enthält den vollständigen Pfad und eine GUID zur eindeutigen Identifizierung.	String	nein
medienFile	URL zur Abholung der Mediadatei Sie enthält den vollständigen Pfad und eine GUID zur eindeutigen Identifizierung.	String	nein
verarbeitungsstatus	Statusbezeichner, um festzulegen, wie der Katalog bearbeitet werden soll (siehe Tabelle 11: Statuscodes der Katalogverarbeitung)	Integer	nein

Rahmenvertrag			
vertragsnummer	Rahmenvertragsnummer des verlinkten Rahmenvertrags	String	nein
rahmenvertragId	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	GUID	nein
Lieferant			
lieferantencockpitId	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	nein

Tabelle 6: Attribut-Definitionen der API „Katalogbereitstellung“

5.2.2. Katalogdatei abholen

Die Abholung von Katalogdateien erfolgt über diesen Endpunkt:

HTTP GET an: API Basis-URL + /cs/rahmenvertrag/kataloge/{freigabeld}

Authorization Header: Bearer + Access Token

Parameter: Freigabeld (GUID als String)

Dieser Endpunkt erfordert eine gültige „Freigabeld“ (GUID als String), um eine Zuordnung zu den Katalogdaten herzustellen. Somit wird über diesen Endpunkt die angebundene Katalogdatei zurückgegeben.

5.2.2.1. Response Codes

Mögliche Response Codes lauten:

Response Code	Mögliche Ursachen
200 OK	Die angeforderte Datei konnte erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiupload
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header

403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
---------------	------------------------------------------------------------------------------------------------------

Tabelle 7: Mögliche Response Codes im Rahmen der Abholung von Katalogdateien

5.2.2.2. Response Payload

Dateiausgabe:

Datei	Dateityp
Katalog	.xml

Tabelle 8: Dateiausgabe der Katalogdatei

5.2.3. Mediadatei abholen

Die Abholung von Mediadateien erfolgt über diesen Endpunkt:

HTTP GET an: API Basis-URL + /cs/rahmenvertrag/medien/{freigabeld}

Authorization Header: Bearer + Access Token

Parameter: freigabeld (GUID als String)

Dieser Endpunkt erfordert eine gültige „Freigabeld“ (GUID als String), um eine Zuordnung zu den Katalogdaten herzustellen. Somit wird über diesen Endpunkt die angebundene Mediadatei zurückgegeben.

5.2.3.1. Response Codes

Mögliche Response Codes lauten:

Response Code	Mögliche Ursachen
200 OK	Die angeforderte Datei konnte erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiupload
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht

Tabelle 9: Mögliche Response Codes im Rahmen der Abholung von Mediadateien

5.2.3.2. Response Payload

Dateiausgabe:

Datei	Dateityp
Mediadatei	.zip

Tabelle 10: Dateiausgabe der Mediadatei

5.2.4. Katalogverarbeitung – Statuscodes

Um dem Katalogsystem mitteilen zu können, wie die Katalogdateien zu verarbeiten sind, werden folgende Statuscodes mitgegeben:

Status	Bezeichnung	Anmerkungen
0	Neuer Katalog	legt den Produktcontainer mit den Artikeln aus dem BMEcat an
1	Ergänzen	fügt ausschließlich Artikel, die nicht im Produktcontainer vorhanden sind, aus dem BMEcat hinzu
2	Aktualisieren	ändert vorhandene Artikel und fügt neue aus dem BMEcat hinzu
3	Überschreiben	ersetzt den Produktcontainer komplett mit den Artikeln aus dem BMEcat

Tabelle 11: Statuscodes der Katalogverarbeitung

6. Statusanpassungen empfangen

Über diese API werden vom Katalogsystem entsprechende Statusanpassungen zu Rahmenvertragsvorgängen an das Lieferantencockpit gesendet. Diese Daten werden den entsprechenden Rahmenvertragsvorgängen zugeordnet und dort verarbeitet.

6.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

6.2. API-Beschreibung

Der folgende Endpunkt dient zum Setzen des Status der Katalogdaten im Lieferantencockpit. Hierzu müssen entsprechende Daten von den Katalogsystemen an diese API übergeben werden.

HTTP POST an: API Basis-URL + cs/rahmenvertrag/statusanpassung

Authorization Header: Bearer + Access Token

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 9: Insomnia-Beispiel der API „Statusanpassungen empfangen“).

6.2.1. POST Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "katalogSystem": {
    "rahmenvertrag": {
      "rahmenvertragId": "ad1a02c7-df04-4fa4-b0e1-209485657881",
      "vertragsnummer": "RV-001-001"
    },
    "katalogfreigabe": {
      "freigabeId": "df69b42b-b6b5-4dc8-9afa-ebd6f43b4586",
      "losId": "0978c1f7-f89b-4e8f-931d-ed43d1500cdb",
      "status": 5
    }
  }
}
```

Codeblock 4: Beispiel - JSON Endpunkt „Statusanpassungen empfangen“

6.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Rahmenvertrag			
vertragsnummer	Rahmenvertragsnummer des verlinkten Rahmenvertrags	String	ja
rahmenvertragsid	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	String	ja
Katalogfreigabe			
freigabelid	eindeutiges Identifikationsmerkmal der Katalogfreigabe Diese GUID ist notwendig, um eine eindeutige Beziehung zwischen Katalogdateien, Mediendateien und Katalogfreigabe herzustellen.	String	nein
losid	eindeutiges Identifikationsmerkmal des Loses Diese GUID ist notwendig, um eine eindeutige Beziehung zwischen Lieferanten, Rahmenvertrag und Los herzustellen.	String	ja
status	siehe Tabelle 14: Mögliche Statusangaben	Integer	ja

Tabelle 12: Attribut-Definitionen der API „Statusanpassung empfangen“

6.4. Response Codes

Mögliche Response Codes lauten:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiuupload 4. asynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header

403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt. Das Format der übermittelten Daten ist nicht mit dem Datenformat des Servers kompatibel.
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 13: Mögliche Response Codes der API „Statusanpassungen empfangen“

6.5. Mögliche Statusangaben

Die API kann die nachfolgenden Status verarbeiten. Status, die nicht über die API gesetzt werden können, werden durch die Anwendungslogik gesetzt. Sie sind der Übersichtlichkeit halber trotzdem hier aufgeführt.

Status	Technischer Bezeichner	Auslöser	Über API setzbar	Optional*	Meldung in UI
0	Katalogdaten fehlen	initialer Status	<input checked="" type="checkbox"/>		Katalogdaten fehlen noch
1	Katalog inaktiv	Änderung durch Behörde initiiert	<input checked="" type="checkbox"/>		Katalog inaktiv
2	Dateien nicht hochgeladen	Datei abgelehnt	<input checked="" type="checkbox"/>		Hochladen fehlgeschlagen
3	Dateien nicht verarbeitet	Datei konnte nicht verarbeitet werden	<input checked="" type="checkbox"/>		Hochladen fehlgeschlagen
4	Dateien zur Abholung bereit	Datei(en) geprüft, valide und hochgeladen	<input checked="" type="checkbox"/>		Katalogdaten bereitgestellt

5	Dateien erfolgreich übermittelt	Katalogsystem hat Dateien vollständig hochgeladen und akzeptiert	☑		Katalogdaten bereitgestellt
6	Dateien nicht akzeptiert	Katalog kann im Katalogsystem nicht verarbeitet werden	☑		Katalogdaten nicht akzeptiert
7	Katalog in Prüfung	Katalog wird durch strategische:n Einkäufer:in geprüft	☑	☑	Vertragspartner prüft Katalogdaten
8	Katalogfreigabe	Katalog wurde durch strategische:n Einkäufer:in freigegeben	☑		Katalogdaten freigegeben
9	Katalog abgelehnt	Katalog wurde durch strategische:n Einkäufer:in abgelehnt	☑		Katalogdaten abgelehnt
10	Änderungen erforderlich	Ein aktiver Katalog benötigt Änderungen	☑		Änderungen stehen an
11	Änderungen nicht hochgeladen	Datei abgelehnt	☒		Hochladen fehlgeschlagen

Tabelle 14: Mögliche Statusangaben

* Optionale Status sind nicht für einen kompletten Workflow (siehe hierzu Abbildung 3: Workflow des Katalogmanagements oder Abbildung 5: Workflow des Katalogmanagements (vergrößerte Darstellung)) erforderlich, sondern dienen lediglich zur Information für Lieferanten.

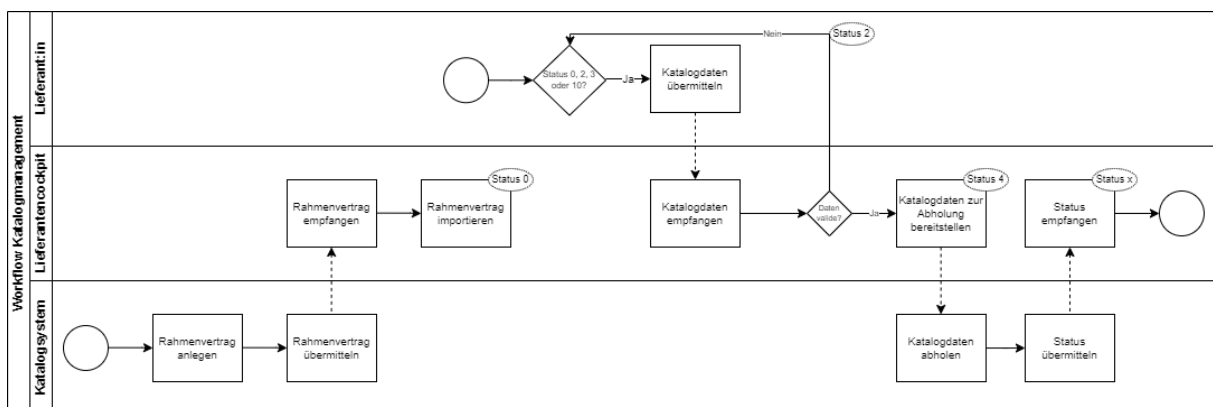


Abbildung 3: Workflow des Katalogmanagements¹

¹ Siehe vergrößerte Darstellung der Grafik im Anhang: Abbildung 5: Workflow des Katalogmanagements (vergrößerte Darstellung).

7. Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen

Über diese API wird vom Katalogsystem der Widerruf der Freischaltungen von Rahmenverträgen an das Lieferantencockpit gesendet. Widerrufene Rahmenverträge stehen den Lieferanten nur noch eingeschränkt zur Verfügung.

7.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

7.2. API-Beschreibung

Der Widerruf der Freischaltung von Rahmenverträgen erfolgt über diesen Endpunkt:

HTTP POST an: API Basis-URL + /cs/rahmenvertrag/freischaltungzurueckziehen

Authorization Header: Bearer + Access Token

7.2.1. POST Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "lieferantencockpitId": "ad1a02c7-df04-4fa4-b0e1-209485657881",
  "rahmenvertragId": "c01a151b-abb7-4aeb-9556-34557c9270aa"
}
```

Codeblock 5: Beispiel - JSON Endpunkt „Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen“

7.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Beratungsanfrage			
lieferantencockpitId	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	ja

rahmenvertragId	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	String	ja
-----------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------	--------	----

Tabelle 15: Attribut-Definitionen der API „Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen“

7.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	<p>Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt.</p> <p>Das Format der übermittelten Daten ist nicht mit Datenformat des Servers kompatibel.</p>
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>

503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.
---------------------------	--------------------------------------------------------------------------------------------------------

Tabelle 16: Mögliche Response Codes der API „Freischaltung des Rahmenvertrags eines Lieferanten zurückziehen“

8. Beratungsanfragen empfangen

Über diese API werden vom Katalogsystem Beratungsanfragen an das Lieferantencockpit gesendet. Diese Daten werden in einer vom Lieferantencockpit betriebenen Datenbank gespeichert und stehen somit den Lieferanten beim Login („Mein Unternehmenskonto“) zur Verfügung.

Alle eingegangenen Beratungsanfragen werden im Status „Antwort fehlt noch“ erstellt. Eine Aktualisierung der eingegangenen Beratungsanfragen ist nicht vorgesehen. Die Warenkorbvorlage (falls vorhanden) orientiert sich am Format „XBestellung“ und wird entsprechend auf das Vorhandensein eines gültigen JSON-Schemas validiert.

8.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

8.2. API-Beschreibung

Das Entgegennehmen von Beratungsanfragen erfolgt über diesen Endpunkt:

HTTP POST an: API Basis-URL + /v1/beratung/erstellen

Authorization Header: Bearer + Access Token

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 10: Insomnia-Beispiel der API „Beratungsanfragen empfangen“).

8.2.1. POST Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "beratungsanfrage": {
    "beratungsanfrageId": "A4GH4KL76GF",
    "datum": "2022-04-13T00:00:00.000Z",
    "dienststelle": "JVA Bremen Oslebshausen",
    "bestellerVorname": "Max",
    "bestellerNachname": "Mustermann",
    "kurzbezeichnung": "Frage zu Bürostuhl",
    "text": "Gibt es den Bürostuhl in Gelb auch mit Lehne?"
  },
  "lieferant": {
    "lieferantencockpitId": "56798a4e-0178-4270-b137-c4ece52bc810",
    "email": "erika.musterfrau@inovation-ag.de"
  },
  "warenkorb": [
    {
      "position": 1,
      "artikelbezeichnung": "Bürostuhl der Firma XY",
      "identifikatorLieferant": "1200120012",
      "menge": "4",
      "merkmale": [
        {
          "eigenschaft": "Farbe",
          "wert": "Schwarz"
        }
      ]
    }
  ]
}
```



```

    {
      "eigenschaft": "Höhe in cm",
      "wert": "120.0"
    }
  ]
}

```

Codeblock 6: Beispiel - JSON Endpunkt „Beratungsanfragen empfangen“

8.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Beratungsanfrage			
beratungsanfragegeld	eindeutiges Identifikationsmerkmal der Beratungsanfrage Diese ID ist notwendig, um einen eindeutigen Bezug zur Beratungsanfrage im Katalogsystem herstellen zu können.	String	ja
datum	Erstellungsdatum der Anfrage	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)	ja
dienststelle	Dienststelle der bestellenden Person	String	ja
bestellerVorname	Vorname der bestellenden Person	String	ja
bestellerNachname	Nachname der bestellenden Person	String	ja
bestellerEmail	E-Mail-Adresse der bestellenden Person	String	nein
kurzbezeichnung	Kurzbezeichnung der Anfrage	String	ja
text	Frage der bestellenden Person	String (max. 2000)	ja
Lieferant			

lieferantencockpitid	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	ja
email	E-Mail-Adresse des Lieferanten	String	ja
Warenkorb		Array (Objekt)	nein
position	Position des Artikels im Warenkorb	Integer	ja
artikelbezeichng	Kurzbeschreibung des Artikels	String	ja
identifikatorLieferant	Eindeutiger Identifikator im Lieferantensystem	String	ja
menge	Anzahl der Artikel	String	ja
Merkmale		Array (Objekt)	nein
eigenschaft	Eigenschaft des Merkmals	String	ja
wert	Wert des Merkmals	String	ja

Tabelle 17: Attribut-Definitionen der API „Beratungsanfragen empfangen“

8.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none">1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)2. abgelaufene Cookies3. zu große Dateien bei Dateiupload4. asynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt. Das Format der übermittelten Daten ist nicht mit Datenformat des Servers kompatibel.
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 18: Mögliche Response Codes der API „Beratungsanfragen empfangen“

9. Beratungsanfragen beantwortet

Über diese Schnittstelle können Katalogsysteme bereits beantwortete Beratungsanfragen abrufen. Eine JSON-Response enthält die bereitgestellten Beratungsdaten und deren zugehörige Warenkorbvorschläge, falls diese vorhanden sind.

9.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

9.2. API-Beschreibung

Diese API dient zur Abholung von freigegebenen Beratungsanfragen und den zugeordneten Warenkörben.

HTTP GET an: API Basis-URL + /v1/beratung/beantwortet

Authorization Header: Bearer + Access Token

Optionale Query-Parameter

Parameter	Wert	Beschreibung	Typ
antwortDatum	2022-04-13T00:00:00.000Z	Einschränkung der Ergebnismenge auf Einträge, deren Datum „antwortDatum“ größer gleich dem angegeben Datum ist	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)

Tabelle 19: Optionale Query-Parameter der API „Beratungsanfragen beantworten“

Beispiel: API Basis-URL + /v1/beratung/beantwortet?antwortDatum=2022-04-13T00:00:00.000Z

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 11: Insomnia-Beispiel der API „Beratungsanfragen beantwortet“).

9.3. Response Codes

Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die angeforderten Daten konnten erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	<ol style="list-style-type: none">1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)2. abgelaufene Cookies

	<p>3. zu große Dateien bei Dateupload</p> <p>4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)</p>
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
406 Method Not Acceptable	Nach der Client-Request ist der Server nicht in der Lage, die vom Client im Accept-Header geforderten Formate zu senden.
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 20: Mögliche Response Codes der API „Beratungsanfragen beantworten“

9.4. Response Payload

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```

{
  "beratungsanfragen": [
    {
      "beratungsanfrageId": "A5GA4KL87KR",
      "datum": "2022-08-10T11:55:01+00:00",
      "dienststelle": "Hochschule XYZ",
      "bestellerVorname": "Jochen",
      "bestellerNachname": "Rumler",
      "bestellerEmail": null,
      "kurzbezeichnung": "Frage zu Bürostuhl aus Katalogsystem",
      "text": "Gibt es den Bürostuhl in Gelb auch mit Lehne?",
      "antwortDatum": "2022-11-11T14:04:26.4513509+00:00",
      "antwortText": "Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.",
      "lieferant": {
        "lieferantencockpitId": "56798a4e-0178-4270-b137-c4ece52bc810"
      }
    }
  ]
}

```

```

    },
    "warenkorb": [
      {
        "position": 1,
        "artikelbezeichnung": "Bürostuhl der Firma XY",
        "identifikator_Lieferant": "1200120012",
        "menge": "4",
        "merkmale": [
          {
            "eigenschaft": "Material",
            "wert": "Netzstoff"
          }
        ]
      }
    ]
  }
]
}

```

Codeblock 7: Response - JSON Endpunkt „Beantwortete Beratungsanfragen bereitstellen“

9.5. Attribut-Definitionen

Die im Datensatz ausgegebenen Werte beziehen sich auf die nachfolgenden Datenfelder. Diese Schnittstelle gibt lediglich Daten zurück. Demnach hat sie keine Pflichtfelder.

Attribut	Beschreibung	Typ
Beratungsanfrage		
beratungsanfrageld	eindeutiges Identifikationsmerkmal der Beratungsanfrage Diese ID ist notwendig, um einen eindeutigen Bezug zur Beratungsanfrage im Katalogsystem herstellen zu können. Bei Beratungsanfragen, die vom Lieferanten angelegt wurden, hat die ID den Wert null.	String
datum	Erstellungsdatum der Anfrage	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)
dienststelle	Dienststelle der bestellenden Person	String
bestellerVorname	Vorname der bestellenden Person	String
bestellerNachname	Nachname der bestellenden Person	String
bestellerEmail	E-Mail-Adresse der bestellenden Person	String

kurzbezeichnung	Kurzbezeichnung der Anfrage	String
text	Frage der bestellenden Person	String
antwortDatum	Zeitpunkt, an dem die Beratungsanfrage beantwortet wurde	DateTime / String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)
antwortText	Antwort des Lieferanten zur angegebenen Beratungsanfrage	String
Lieferant		
lieferantencockpitid	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID
Warenkorb		Array (Objekt)
position	Position des Artikels im Warenkorb	String
artikelbezeichnung	Kurzbeschreibung des Artikels	String
identifikatorLieferant	Eindeutiger Identifikator im Lieferantensystem	String
menge	Anzahl der Artikel	String
Merkmale		Array (Objekt)
eigenschaft	Eigenschaft des Merkmals	String
wert	Wert des Merkmals	String

Tabelle 21: Attribut-Definitionen der API „Beratungsanfragen beantworten“

10. Geschlossene Beratungsanfragen empfangen

Über diese API werden vom Katalogsystem Beratungsanfragen an das Lieferant Cockpit gesendet. Diese Daten werden in einer vom Lieferant Cockpit betriebenen Datenbank gespeichert und stehen somit den Lieferanten beim Login („Mein Unternehmenskonto“) bereit.

Die Aktualisierung von bereits vorhandenen Beratungsanfragen wird durch Abgleich der ID ermöglicht. Falls keine Zuordnung stattfinden kann, wird die eingegangene Anfrage vom System ignoriert. Alle eingegangenen Beratungsanfragen werden in den Status „Anfrage geschlossen“ gesetzt.

10.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

10.2. API-Beschreibung

Das Entgegennehmen von Beratungsanfragen erfolgt über diesen Endpunkt:

HTTP POST an: API Basis-URL + /beratung/schliessen

Authorization Header: Bearer + Access Token

Hinweis: Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 12: Insomnia-Beispiel der API „Geschlossene Beratungsanfragen empfangen“).

10.2.1. POST Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "beratungsanfrage": {
    "beratungsanfrageId": "A4GH4KL76GF",
    "datum": "2022-04-13T00:00:00.000Z",
    "text": "Begründung zur Schließung"
  }
}
```

Codeblock 8: Beispiel - JSON Endpunkt „Geschlossene Beratungsanfragen empfangen“

10.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Beratungsanfrage			
beratungsanfrageld	eindeutiges Identifikationsmerkmal der Beratungsanfrage Diese ID ist notwendig, um einen eindeutigen Bezug zur Beratungsanfrage im Katalogsystem herstellen zu können.	String	ja
Datum	Erstellungsdatum der Anfrage	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)	ja
Text	Begründung zur Schließung der Beratungsanfrage	String (max. 2000)	ja

Tabelle 22: Attribut-Definitionen der API „Geschlossene Beratungsanfragen empfangen“

10.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiuupload 4. asynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header

403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	Nach der Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt. Das Format der übermittelten Daten ist nicht mit dem Datenformat des Servers kompatibel.
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 23: Mögliche Response Codes der API „Geschlossene Beratungsanfragen empfangen“

11. Geschlossene Beratungsanfrage bereitstellen

Es wird eine API erstellt, die Katalogsystemen zur Verfügung gestellt wird. Über diese API können Katalogsysteme Beratungsanfragen abrufen, die im Katalogsystem erstellt und innerhalb des Lieferantcockpits geschlossen wurden. Eine JSON-Response enthält die bereitgestellten Beratungsdaten. Alle Anfragen werden immer im Status „Anfrage geschlossen“ bereitgestellt.

11.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

11.2. API-Beschreibung

Diese API dient zur Abholung von freigegebenen Beratungsanfragen und den zugeordneten Warenkörben.

HTTP GET an: API Basis-URL + /beratung/geschlossen

Authorization Header: Bearer + Access Token

Optionale Query-Parameter

Parameter	Wert	Beschreibung	Typ
datum	2022-04-13T00:00:00.000Z	Einschränkung der Ergebnismenge auf Einträge, deren Datum „datum“ größer gleich dem angegebenen Datum ist	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)

Tabelle 24: Optionale Query-Parameter der API „Geschlossene Beratungsanfragen bereitstellen“

Beispiel: API Basis-URL + /beratung/geschlossen?antwortDatum=2022-04-13T00:00:00.000Z

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 13: Insomnia-Beispiel der API „Geschlossene Beratungsanfragen bereitstellen“).

11.3. Response Codes

Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die angeforderten Daten konnten erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)

	<ol style="list-style-type: none"> 2. abgelaufene Cookies 3. zu große Dateien bei Dateiuupload 4. asynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
406 Method Not Acceptable	Nach der Client-Request ist der Server nicht in der Lage, die vom Client im Accept-Header geforderten Formate zu senden.
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 25: Mögliche Response Codes der API „Geschlossene Beratungsanfragen bereitstellen“

11.4. Response Payload

Der ausgegebene Datensatz entspricht folgender Struktur (Beispiel):

```

{
  "beratungsanfragen": [
    {
      "beratungsanfrageId": "A4GH4KL76GF",
      "datum": "2022-04-13T00:00:00.000Z",
      "text": "Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam
erat, sed diam voluptua."
    }
  ]
}

```

Codeblock 9: Response - JSON Endpunkt „Geschlossene Beratungsanfragen bereitstellen“

11.5. Attribut-Definitionen

Die im Datensatz ausgegebenen Werte beziehen sich auf die nachfolgenden Datenfelder. Diese API gibt lediglich Daten zurück. Demnach hat sie keine Pflichtfelder.

Attribut	Beschreibung	Typ
Beratungsanfrage		
beratungsanfrageld	eindeutiges Identifikationsmerkmal der Beratungsanfrage Diese ID ist notwendig, um einen eindeutigen Bezug zur Beratungsanfrage im Katalogsystem herstellen zu können.	String
datum	Erstellungsdatum der Anfrage	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)
text	Begründung zur Schließung der Beratungsanfrage Wenn keine Eingabe durch die Lieferanten erfolgt, wird automatisch "Geschlossen durch Lieferanten" übermittelt.	String

Tabelle 26: Attribut-Definitionen der API „Geschlossene Beratungsanfragen bereitstellen“

12. Bestellung bereitstellen

Über diese API können Katalogsysteme Informationen zu den Bestellaufträgen abrufen, die innerhalb des Lieferantencockpits verarbeitet wurden.

Eine JSON-Response enthält die bereitgestellten Bestelldaten sowie den aktuellen Status des Bestellauftrags.

12.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

12.2. API-Beschreibung

Diese API dient zur Abholung von Bestellaufträgen, deren Status und zugeordneten Anhängen.

HTTP GET an: API Basis-URL + /bestellung/bereitstellung

Authorization Header: Bearer + Access Token

Optionale Query-Parameter

Parameter	Wert	Beschreibung	Typ
datum	2022-04-13T00:00:00.000Z	Einschränkung der Ergebnismenge auf Einträge, deren Datum „datum“ größer gleich dem angegebenen Datum ist	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)

Tabelle 27: Optionale Query-Parameter der API „Bestellung bereitstellen“

Beispiel: API Basis-URL + /bestellung/bereitstellung?datum=2022-04-13T00:00:00.000Z

Hinweis: Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 14: Insomnia-Beispiel der API „Bestellung bereitstellen“).

12.3. Response Codes

Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die angeforderten Daten konnten erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)

	<ul style="list-style-type: none"> 2. abgelaufene Cookies 3. zu große Dateien bei Dateiupload 4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
406 Method Not Acceptable	Nach der Client-Request ist der Server nicht in der Lage, die vom Client im Accept-Header geforderten Formate zu senden.
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 28: Mögliche Response Codes der API „Bestellung bereitstellen“

12.4. Response Payload

Der ausgegebene Datensatz entspricht folgender Struktur (Beispiel):

```
{
  "bestellauftrag": [
    {
      "bestellungId": "4100001234",
      "datum": "2022-05-24T00:00:00.000Z",
      "text": "Bestellung nicht lieferbar",
      "status": "2",
      "anhaenge": [
        "https://api-stage.lieferantcockpit.de/bestellung/anhaenge/72b2b42f-6a4e-4a5b-9b0d-2124ca49e388",
        "https://api-stage.lieferantcockpit.de/bestellung/anhaenge/2770c3ae-23a8-4063-ab34-eaefa814088d"
      ]
    }
  ]
}
```

Codeblock 10: Response - JSON Endpunkt „Bestellung bereitstellen“

12.5. Attribut-Definitionen

Die im Datensatz ausgegebenen Werte beziehen sich auf die nachfolgenden Datenfelder. Diese API gibt lediglich Daten zurück. Demnach hat sie keine Pflichtfelder.

Attribut	Beschreibung	Typ
Beratungsanfrage		
bestellungId	eindeutiges Identifikationsmerkmal der Bestellung Diese ID ist notwendig, um einen eindeutigen Bezug zur Bestellung im Katalogsystem herstellen zu können.	String
datum	Datum der letzten Statusänderung	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)
text	Anmerkung zur letzten Statusanpassung	String
status	aktueller Status des Bestellauftrags	String

	siehe Tabelle 31: Mögliche Statusmeldungen der API „Bestellung“	
Anhänge		
	Array mit URLs zur Abholung der Anhänge Es enthält jeweils den vollständigen Pfad und eine GUID zur eindeutigen Identifizierung.	String-Array

Tabelle 29: Attribut-Definitionen der API „Bestellung bereitstellen“

12.5.1. Anhang zum Bestellauftrag abholen

Diese API dient zur Abholung von Anhängen der Bestellaufträge und erfordert eine gültige Anhang-ID (GUID als String), um eine Zuordnung zur Anhang-Datei herzustellen. Somit wird über diesen API-Endpunkt der entsprechende Anhang zurückgegeben.

HTTP GET an: API Basis-URL + /bestellung/anhaenge/{AnhangId}

Parameter: AnhangId (GUID als String)

Authorization Header: Bearer + Access

Mögliche Response Codes lauten:

Response Code
200 OK
400 Bad Request
401 Unauthorized
403 Forbidden

Tabelle 30: Mögliche Response Codes der API „Bestellung bereitstellen“ - Bestellung abholen

Der Header Content-Disposition enthält die Informationen zum Dateityp und den Dateinamen. Letzterer setzt sich aus dem Präfix "OrderAttachment" und der Anhang-ID zusammen.

12.5.2. Response Payload

Der Payload der Response besteht ausschließlich aus der Datei.

12.6. Mögliche Statusangaben

Die API gibt folgende Statusangaben zu Bestellungen aus:

Status	Technische Bezeichner	Auslöser	Content (Meldung im UI)
0	Bestellung offen	Initialer Status	Neue Bestellung
1	Bestellung in Bearbeitung	In Bearbeitung durch Lieferanten (Bestell-auftrag wurde geöffnet)	Angaben fehlen noch
2	Bestellung lieferbar	Rückmeldung von dem Lieferanten	Angaben fehlen noch
3	Bestellung nicht lieferbar	Rückmeldung von dem Lieferanten im Falle einer nicht lieferbaren Bestellung	Rückmeldung erwartet
4	Bestellung storniert	Bestellung durch Besteller:in oder strat. Einkäufer:in storniert (Status wird durch Katalogsystem gesetzt)	Bestellung storniert
5	Lieferung versendet	Zustellung wurde veranlasst	Auftrag erfüllt

Tabelle 31: Mögliche Statusmeldungen der API „Bestellung bereitstellen“

13. Bestellung empfangen

Über diese API werden vom Katalogsystem Bestellaufträge an das Lieferantencockpit gesendet. Diese Daten werden in einer vom Lieferantencockpit betriebenen Datenbank gespeichert und stehen somit den Lieferanten beim Login („Mein Unternehmenskonto“) zur Verfügung.

Alle eingegangenen Bestellaufträge werden mit dem initialen Status vom Lieferantencockpit entgegengenommen.

13.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

13.2. API-Beschreibung

Das Entgegennehmen von Bestellungen erfolgt über diesen Endpunkt.

HTTP POST an: API Basis-URL + /v2/bestellung/empfangen

POST Header:

Authorization Header: Bearer + Access Token

Content-Type: multipart/form-data – Notwendig, um sowohl die Metadaten als auch die XBestellung-Dateien zu übermitteln.

Der Endpunkt erfordert eine Struktur, die in zwei Bereiche unterteilt werden kann. Die Metadaten dienen der Zuordnung der Daten zum Lieferanten und konfigurieren die anzuwendende Verarbeitungslogik. Die Bestellung enthält die Daten zum Bestellauftrag in strukturierter Form. Das Lieferantencockpit gibt vor, welche Formate akzeptiert und verarbeitet werden. Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 15: Insomnia-Beispiel der API „Bestellung empfangen“).

13.2.1. Request (Teil Metadaten)

Die Schnittstelle erfordert für den Teil Metadaten eine folgende Struktur (Beispiel):

```
{
  "lieferant": {
    "lieferantcockpitId": "56798a4e-0178-4270-b137-c4ece52bc810",
    "lieferantemail": "erika.musterfrau@innovation-ag.de"
  },
  "mapping": {
    "rahmenvertrag": [
      {
        "rahmenvertragid": "ad1a02c7-df04-4fa4-b0e1-209485657881",
        "positionsnummer": "0010"
      },
      {
        "rahmenvertragid": "4bef66cf-ee6d-4e9f-b73c-4bb2533939f5",
        "positionsnummer": "0020"
      },
      {
        "rahmenvertragid": "e2de5857-cada-4891-b194-219b18cd19b6",
        "positionsnummer": "0030"
      }
    ]
  },
  "konfiguration": {
    "dateiformat": "xbestellung",
    "nurvalidieren": false
  }
}
```

Codeblock 11: Beispiel - JSON Endpunkt „Bestellungen empfangen“ - Teil Metadaten

13.2.1.1. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Lieferant			
lieferantcockpitid	eindeutiges Identifikationsmerkmal des Lieferanten im Katalogsystem Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	ja

Lieferantemail	E-Mail-Adresse des Lieferanten/Empfängers	String	ja
Mapping			nein
Rahmenvertrag			nein
rahmenvertragsid	externer Identifier (GUID) zur eindeutigen Zuordnung der Position aus dem Bestellauftrag zum hinterlegten Rahmenvertrag	String	ja
positionsnummer	"Line item identifier" aus der XBestellung Es wird überprüft, ob ein Line Item Identifier mit dem angegebenen Wert in der XBestellung vorhanden ist. Daneben wird ein Bezug von Position zu Rahmenvertrag hergestellt.	String	ja
Bestellung			
Dateiformat	gibt das Format der angefügten Datei an Dieses bestimmt, wie die eingehende Datei durch die Schnittstelle verarbeitet wird. Erlaubte Werte sind: <ul style="list-style-type: none">• XBestellung• JSON Weitere Formate können in zukünftigen Releases hinzugefügt werden.	String	ja
Nurvalidieren	legt fest, ob die übermittelte XBestellung nur validiert werden soll oder ob sie anschließend ebenfalls als Bestellauftrag angelegt und versendet wird Erlaubte Werte sind true und false. Default: false	Boolean	Nein

Tabelle 32: Attribut-Definitionen der API „Bestellung empfangen“ - Teil Metadaten

13.2.2. Request (Teil Bestellung)

Die Schnittstelle erfordert für den Teil Bestellung eine Datei in einem der erlaubten Formate. Das Dateiformat „XBestellung“ richtet sich zum jetzigen Zeitpunkt nach der Spezifikation XBestellung (v0.8) und wird dahingehend validiert.

Das Dateiformat „JSON“ orientiert sich am Modul „XBestellung generieren“ und wird im folgenden Format erwartet:

```
{
  "bestellung": {
    "bestellnummer": "4100001234",
    "waehrung": "EUR",
    "bestelldatum": "2021-02-05",
    "enddatum": "2021-02-05",
    "kommentar": "Dies ist ein nur eine Testbestellung.",
    "besteller": "Mark Hapke",
    "positionen": [
      {
        "kurzbeschreibung": "Ein gelber Bürostuhl",
        "langbeschreibung": "eleganter Stuhl mit komfortablen Armleh-
nen...",
        "artikelnummer": "01200120012",
        "eclass_id": "34234534",
        "eclass_verzeichnisname": "CD-, DVD-Aufbewahrung",
        "menge": 10,
        "bestelleinheit": "C62",
        "nettogesamtpreis": 100.0,
        "waehrung": "EUR",
        "nettopreis": 10.0,
        "preismenge": 1,
        "steuer_code": "S",
        "steuersatz": 19,
        "steuerart": "VAT",
        "teillieferung": "false",
        "positionsnummer": "0010",
        "finanzposition": "2342345245245245",
        "merkmale": [
          {
            "merkmal": "Farbe der Armlehne",
            "wert": "gelb"
          },
          {
            "merkmal": "Farbe der Rückenlehne",
            "wert": "gelb"
          }
        ]
      }
    ]
  },
  "bestellerdaten": {
    "leitweg_id": "04011000-900X01-85",
    "titel": "JVA Bremen Oslebshausen",
    "zusatz": "",
    "stadt": "Bremen",
    "plz": "28239",
  }
}
```

```

        "strasse_hausnummer": "Am Fuchsberg 3",
        "land": "Deutschland",
        "ansprechpartner": "Andree Holstenheim",
        "telefon": "0421 3616279",
        "email": "office@jva.bremen.de"
    },
    "rechnungsdaten": {
        "peppol_participant_id": "04011000-900X01-85",
        "leitweg_id": "04011000-900X01-85",
        "titel": "JVA Bremen Oslebshausen",
        "zusatz": "Werkstatt",
        "stadt": "Bremen",
        "plz": "28239",
        "strasse_hausnummer": "Am Fuchsberg 3",
        "land": "Deutschland",
        "ansprechpartner": "Andree Holstenheim",
        "telefon": "0421 3616279",
        "email": "office@jva.bremen.de"
    },
    "lieferdaten": {
        "titel": "JVA Bremen Oslebshausen",
        "zusatz": "Werkstatt",
        "stadt": "Bremen",
        "plz": "28239",
        "strasse_hausnummer": "Am Fuchsberg 3",
        "land": "Deutschland",
        "lieferhinweis": "an der Pforte klingeln",
        "ansprechpartner": "Werkstattleiter Öhrsen",
        "telefon": "0421 3616282",
        "email": "werkstatt@jva.bremen.de"
    },
    "lieferantendaten": {
        "kreditorennummer": "122000231",
        "titel": "Tolle Büromöbel GmbH",
        "zusatz": "",
        "stadt": "Lilienthal",
        "plz": "28865",
        "strasse_hausnummer": "Trupe 6",
        "land": "Deutschland",
        "ansprechpartner": "Frau Trovelt"
    }
}
}

```

Codeblock 12: Beispiel - JSON Endpunkt „Bestellungen empfangen“ - Teil Bestellung

13.2.2.1. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Referenz (ID in Standard XBestellung)	Pflichtfeld
Lieferant				
lieferantcockpitid	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID		ja
lieferantemail	E-Mail-Adresse des Lieferanten/Empfängers	String		ja
Bestellung				
bestellnummer	Order identifier	String	BT-T01-3	ja
waehrung	Currency	String (Code: DocumentCurrencyCode)	BT-T01-9	ja
bestelldatum	Order issue date	Date (Format yyyy-mm-dd)	BT-T01-5	ja
enddatum	Order validity end date	Date (Format yyyy-mm-dd)	BT-T01-12_1	nein
kommentar	Document level textual note	String	BT-T01-8	nein
besteller	Buyer contact	String	BT-T01-10	nein
Bestellerdaten (BG-T01-4Pflicht)				
leitweg_id	Buyer party electronic address	String	BT-T01-21	ja

titel	Buyer name / Buyers legal registration name / Originator name	String	BT-T01-23_1, BT-T01-33, BT-T01-58_1	ja
zusatz	Address line 2	String	BT-T01-25	nein
stadt	City / Buyers legal registration address city name	String	BT-T01-26, BT-T01-35	nein
plz	Post code	String	BT-T01-27	nein
strasse_hausnummer	Address line 1	String	BT-T01-24	nein
land	Country	String	BT-T01-30	ja
ansprechpartner	Contact person name	String	BT-T01-59	nein
telefon	Contact telephone number	String	BT-T01-60	nein
email	Contact email address	String	BT-T01-61	nein
Rechnungsdaten (BG-T01-20, Pflicht)				
peppol_participant_id	Invoicee party electronic address	String	BT-T01-62	nein
titel	Invoicee identifier	String	BT-T01-63_1	ja
zusatz	Invoicee name	String	BT-T01-64_1	nein
stadt	City / City name	String	BT-T01-67, BT-T01-76	nein
plz	Post code	String	BT-T01-68	nein
strasse_hausnummer	Address line 1	String	BT-T01-65	nein
land	Country	String	BT-T01-71	ja
ansprechpartner	Contact person name	String	BT-T01-78	nein
telefon	Contact telephone number	String	BT-T01-79	nein

email	Contact email address	String	BT-T01-80	nein
Lieferdaten				
titel	Delivery location name / Delivery party name	String	BT-T01-82, BT-T01-93_1	ja
zusatz	Address line 2	String	BT-T01-84	nein
stadt	City	String	BT-T01-85	nein
plz	Post code	String	BT-T01-86	nein
strasse_hausnummer	Address line 1	String	BT-T01-83	nein
land	Country	String	BT-T01-89	ja
lieferhinweis	Address line 3	String	BT-T01-88_1	nein
ansprechpartner	Contact person name	String	BT-T01-94	nein
telefon	Contact telephone number	String	BT-T01-95	nein
email	Contact email address	String	BT-T01-96	nein
Lieferantendaten (BG-T01-11, Pflicht)				
kreditorennummer	Seller party electronic address	String	BT-T01-40	ja
titel	Seller party trading name / Seller legal registration name	String	BT-T01-42_1, BT-T01-50	ja
zusatz	Address line 2	String	BT-T01-44	nein
stadt	City / Sellers legal registration address city name	String	BT-T01-45, BT-T01-52	nein
plz	Post code	String	BT-T01-46	nein
strasse_hausnummer	Address line 1	String	BT-T01-43	nein
land	Country	String	BT-T01-49	ja

ansprechpartner	Contact person name	String	BT-T01-54	nein
Positionen				
kurzbeschrei- bung	Item name	String	BT-T01-148	ja
langbeschrei- bung	Item description	String	BT-T01-147	nein
artikelnummer	Item standard identifier	String	BT-T01-151_1	nein
eclass_id	Item classification code	String	BT-T01-153_1	nein
eclass_verzeich- nisname	Clear text name equivalent of classification code	String	BT-T01-153_4	nein
menge	Ordered quantity	Decimal	BT-T01-128	ja
bestelleinheit	Ordered quantity unit of measure	String (Code: UnitCode)	BT-T01-128_1, BT-T01-143_1	ja
netto- gesamtpreis	Order line amount	Decimal	BT-T01-129	nein
waehrung	Currency identifier	String (Code: UnitCode)	BT-T01-129_1, BT-T01-142_1	ja
nettopreis	Item price	Decimal	BT-T01-142	ja
preismenge	Item price base quantity	Decimal	BT-T01-143	nein
steuer_code	Item TAX category code	String (Code: UNCL5305)	BT-T01-154	ja
steuersatz	Line TAX rate	Decimal	BT-T01-155	nein
steuerart	Tax scheme identifier	String	BT-T01-156_1	ja
teillieferung	Partial delivery indicator	Boolean	BT-T01-130	nein
positionsnum- mer	Line item identifier	String	BT-T01-127	ja

finanzposition	Buyer contact	String	BT-T01-10	nein
Merkmale				
merkmal	Item property name	String	BT-T01-157	ja
wert	Item property value	String	BT-T01-159	ja

Tabelle 33: Attribut-Definitionen der API „Bestellung empfangen“ - Teil Bestellung

13.2.3. Response Codes

Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die Validierung der XBestellung war erfolgreich. Es wurde kein Bestellauftrag angelegt.
201 Created	Die Generierung und Validierung der XBestellung war erfolgreich. Es wurde ein Bestellauftrag angelegt.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiuupload 4. asynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein) 5. ungültig oder nicht validierbare Bestellung übermittelt 6. Bestelldaten nicht vollständig (z. B. E-Mail-Adresse der empfangsberechtigten Person nicht vorhanden)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt. Das Format der übermittelten Daten ist nicht mit Datenformat des Servers kompatibel.
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen

	Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 34: Mögliche Response Codes der API „Bestellung empfangen“

13.3. Response Payload

Ein Response-Payload ist nur im Fall einer nicht validierbaren Bestellung vorhanden. In diesem Fall wird ein im Validator erstellter Prüfbericht mit einem Statuscode als Response zurückgegeben. Dieser Bericht kann zur Einsicht durch das nachnutzende Katalogsystem dargestellt oder als HTML-Datei abgespeichert werden.

Der ausgegebene Datensatz entspricht folgender Struktur (verkürztes Beispiel):

```

{
  "statusCode": 406,
  "validationReport": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><rep:re-
report xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:rep=\"http://www.xoev.de/de/validator/var1/1\"
xmlns:s=\"http://www.xoev.de/de/validator/framework/1/scenarios\"
xmlns:in=\"http://www.xoev.de/de/validator/framework/1/createreportinput\"
xmlns:svrl=\"http://purl.oclc.org/dsdl/svrl\" xmlns:xd=\"http://www.oxygen-
xml.com/ns/doc/xsl\" [...]"
}

```

Codeblock 13: Beispiel - Response Payload „Bestellung empfangen“

14. Bestellung stornieren

Über diese API werden vom Katalogsystem Bestellaufträge im Lieferantencockpit storniert. Diese Daten werden in einer vom Lieferantencockpit betriebenen Datenbank aktualisiert und stehen somit den Lieferanten beim Login („Mein Unternehmenskonto“) bereit.

14.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

14.2. API-Beschreibung

Die Stornierung eines Bestellauftrags erfolgt über diesen Endpunkt.

HTTP POST an: API Basis-URL + /bestellung/stornieren

Authorization Header: Bearer + Access Token

14.2.1. Request Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "bestellungId": "4100001234",
  "lieferantEmail": "erika.musterfrau@innovation-ag.de"
}
```

Codeblock 14: Beispiel - JSON Endpunkt „Bestellung stornieren“

Die Bildschirmaufnahme eines Insomnia-Beispiels finden Sie im Anhang (siehe hierzu Abbildung 16: Insomnia-Beispiel der API „Bestellung stornieren“).

14.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
bestellungId	eindeutiges Identifikationsmerkmal der Bestellung Diese ID ist notwendig, um einen eindeutigen Bezug zur Bestellung im Katalogsystem herstellen zu können.	String	ja
lieferantEmail	E-Mail-Adresse des Lieferanten, damit dieser über die Stornierung informiert werden kann	String	ja

Tabelle 35: Attribut-Definitionen der API „Bestellung stornieren“

14.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none">1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)2. abgelaufene Cookies3. zu große Dateien bei Dateiupload4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File

415 Unsupported Media Type	<p>Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt.</p> <p>Das Format der übermittelten Daten ist nicht mit Datenformat des Servers kompatibel.</p>
500 Internal Server Error	<p>generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen</p> <p>Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.</p>
503 Service Not Available	<p>Der Server ist nicht erreichbar, da er aufgrund Wartungsarbeiten abgeschaltet oder überlastet ist.</p>

Tabelle 36: Mögliche Response Codes der API „Bestellung stornieren“

15. Vertragsstörungsmeldung (VSM) empfangen

Über diese API werden vom Katalogsystem Vertragsstörungsmeldungen an das Lieferantencockpit gesendet. Diese Daten werden in einer vom Lieferantencockpit betriebenen Datenbank gespeichert und stehen somit den Lieferanten beim Login („Mein Unternehmenskonto“) zur Verfügung. Alle eingegangenen Vertragsstörungsmeldungen werden im Status „Empfangen“ erstellt.

15.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

15.2. API-Beschreibung

Das Entgegennehmen von Vertragsstörungsmeldungen erfolgt über diesen Endpunkt.

HTTP POST an: API Basis-URL + /v1/vsm/empfangen

Authorization Header: Bearer + Access Token

15.2.1. Request Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
{
  "lieferant": {
    "lieferantencockpitId": "56798a4e-0178-4270-b137-c4ecec52bc810",
    "email": "erika.musterfrau@inovation-ag.de"
  },
  "vertragsstoerungsmeldung": {
    "rahmenvertragid": "8773b405-bbe0-4564-ae2d-c2083c49d8d1",
    "kurzbeschreibung": "Meldung einer Vertragsstörung",
    "meldung": "Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam e-
rat, sed diam voluptua..."
  },
  "absender": {
    "dienststelle": "Behörde XYZ",
    "name": "Max Mustermann",
    "email": "max.mustermann@behörde-xyz.de",
    "telefon": "0000-00000000"
  },
  "anhaenge": [
    "def1853a4a72464eb8b0d357c4f6e8a5",
    "78a0fbb13b2e41c3bc3234259d02c952",
    "1c64598388264936b71f7f193da26d68"
  ]
}
```

Codeblock 15: Beispiel - JSON Endpunkt „VSM empfangen“

15.3. Attribut-Definitionen

Die im Datensatz erforderlichen Werte beziehen sich auf folgende Datenfelder:

Attribut	Beschreibung	Typ	Pflichtfeld
Lieferant			
lieferantencockpitid	eindeutiges Identifikationsmerkmal des Lieferanten im Lieferantencockpit Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID	ja
email	E-Mail-Adresse des Lieferanten	String	ja
Vertragsstoerungsmeldung			
rahmenvertragid	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	GUID	ja
kurzbeschreibung	kurze Beschreibung der Vertragsstörungsmeldung	String (max. 40 Zeichen)	ja
meldung	Beschreibung der Vertragsstörungsmeldung	String (max. 2000 Zeichen)	ja
Absender			
dienststelle	Zuordnung der Vertragsstörungsmeldung zu einer Behörde / dem:der strategischen Einkäufer:in der Behörde	String	nein
name	Vor- und Nachname der Ansprechperson (ggf. strategische:r Einkäufer:in) Kommunikation zwischen Lieferanten und Ansprechpersonen außerhalb des Lieferantencockpits	String	nein

email	E-Mail-Adresse der Ansprechperson (ggf. strategische:r Einkäufer:in) Kommunikation zwischen Lieferanten und Ansprechpersonen außerhalb des Lieferantencockpits	String	nein
telefon	Telefonnummer der Ansprechperson (ggf. strategische:r Einkäufer:in) Kommunikation zwischen Lieferanten und Ansprechpersonen außerhalb des Lieferantencockpits	String	nein
Anhaenge			
	Auflistung der IDs von zuvor hochgeladenen Anhängen, die dem Datensatz zugeordnet werden sollen	String-Array	nein

Tabelle 37: Attribut-Definitionen der API „VSM empfangen“

15.4. Response

Es ist kein Payload vorgesehen. Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die erforderlichen Daten wurden korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none"> 1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL) 2. abgelaufene Cookies 3. zu große Dateien bei Dateiupload 4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen

404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt. Das Format der übermittelten Daten ist nicht mit Datenformat des Servers kompatibel
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 38: Mögliche Response Codes der API „VSM empfangen“

16. Vertragsstörungsmeldung (VSM) Anhang empfangen

Über diese API werden vom Katalogsystem Anhänge zu Vertragsstörungsmeldungen an das Lieferantencockpit gesendet. Der Rückgabewert muss dem Request Body bei der Anlage einer Vertragsstörungsmeldung hinzugefügt werden.

Übermittelte Anhänge werden automatisch nach 7 Tagen gelöscht, wenn diese keiner Vertragsstörungsmeldung zugeordnet werden.

16.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

16.2. API-Beschreibung

Das Entgegennehmen von Anhängen zu Vertragsstörungsmeldungen erfolgt über diesen Endpunkt.

HTTP POST an: API Basis-URL + /v1/vsm/anhaenge/empfangen

Authorization Header: Bearer + Access Token

Content-Type: multipart/form-data

16.2.1. Request Body

Der Endpunkt erfordert eine übergebene Struktur wie folgt (Beispiel):

```
-----974767299852498929531610575
Content-Disposition: form-data; name="attachment"; filename="beispiel.pdf"
Content-Type: application/pdf
-----974767299852498929531610575--
```

Codeblock 16: Beispiel - JSON Endpunkt „VSM Anhang empfangen“

16.3. Response Codes

Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die Datei wurde korrekt übermittelt und der Vorgang konnte erfolgreich durchgeführt werden.
400 Bad Request	<ol style="list-style-type: none">1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)2. abgelaufene Cookies3. zu große Dateien bei Dateiuupload

401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
415 Unsupported Media Type	Nach Client-Request wird dem Server eine Ressource übermittelt, die in einem falschen Medientyp vorliegt. Das Format der übermittelten Daten ist nicht mit Datenformat des Servers kompatibel
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 39: Mögliche Response Codes der API „VSM Anhang empfangen“

16.3.1. Response Payload

Es wird eine eindeutige ID zurückgegeben, die bei der Anlage der VSM angegeben werden muss, um eine Zuordnung des Anhangs zur VSM zu erhalten.

```
{
  "id": "def1853a4a72464eb8b0d357c4f6e8a5"
}
```

Codeblock 17: Beispiel - JSON Response der API „VSM Anhang empfangen“

17. Vertragsstörungsmeldung bereitstellen

Über diese API können Katalogsysteme Vertragsstörungsmeldungen (VSM) abrufen, die im Lieferantencockpit erstellt wurden.

Eine JSON-Response enthält die bereitgestellten Daten zur VSM für das anfragende Katalogsystem.

17.1. Sichere Kommunikation

Details zur sicheren Kommunikation sind zentral im Kapitel 2 Sichere Kommunikation aufgeführt.

17.2. API-Beschreibung

Diese API dient zur Abholung von Vertragsstörungsmeldungen, deren Status und zugeordneten Anhängen.

HTTP GET an: API Basis-URL + /v1/vsm/bereitstellung

Authorization Header: Bearer + Access Token

Optionale Query-Parameter

Parameter	Wert	Beschreibung	Typ
Datum	2022-04-13T00:00:00.000Z	Einschränkung der Ergebnismenge auf Einträge, deren letztes Aktualisierungsdatum größer gleich dem angegeben Datum entspricht	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)

Tabelle 40: Optionale Query-Parameter der API „VSM bereitstellen“

Beispiel: API Basis-URL + /v1/vsm/bereitstellung?datum=2022-04-13T00:00:00.000Z

17.3. Response Codes

Mögliche Response Codes:

Response Code	Mögliche Ursachen
200 OK	Die angeforderten Daten konnten erfolgreich ermittelt und bereitgestellt werden.
400 Bad Request	<ol style="list-style-type: none">1. syntaktischer Fehler in der aufgerufenen URL (fehlerhafte URL, ungültige Zeichen in der URL)2. abgelaufene Cookies3. zu große Dateien bei Dateiuupload

	4. unsynchroner DNS Cache (lokal gespeicherte DNS Daten stimmen nicht mit denen der aufgerufenen Webseite überein)
401 Unauthorized	fehlerhafte oder syntaktisch falsche Angaben im Authentication Header
403 Forbidden	fehlerhafte Konfiguration im .htaccess File oder fehlende Berechtigung, um auf Ressource zuzugreifen
404 Not Found	fehlerhafte Eingabe der URL oder die gewünschte Seite wurde verschoben bzw. gelöscht
405 Method Not Allowed	fehlerhafte Eingabe der URL oder serverseitig fehlerhafte Konfiguration im .htaccess File
406 Method Not Acceptable	Nach der Client-Request ist der Server nicht in der Lage, die vom Client im Accept-Header geforderten Formate zu senden.
500 Internal Server Error	generischer Statuscode, der indiziert, dass der Server daran gehindert wurde, die Client-Anfrage zu erfüllen Die Ursache liegt hier meist bei einem Fehler innerhalb der Applikation.
503 Service Not Available	Der Server ist nicht erreichbar, da er aufgrund von Wartungsarbeiten abgeschaltet oder überlastet ist.

Tabelle 41: Mögliche Response Codes der API „Vertragsstörungsmeldung bereitstellen“

17.4. Response Payload

Der ausgegebene Datensatz entspricht folgender Struktur (Beispiel):

```

{
  "vertragsstoerungsmeldungen": [
    {
      "lieferant": {
        "email": "erika.musterfrau@inovation-ag.de",
        "firmenname": "Inovation AG",
        "lieferantencockpitId": "7135c0df-b741-4929-b699-
a7c44de49f82",
        "vorname": "Erika",
        "nachname": "Musterfrau",
        "telefonnummer": "0211-1234567"
      },
      "rahmenvertragid": "ID90003212312",
    }
  ]
}

```



```

        "erstelldatum": "2020-07-01T14:30:00.001Z",
        "kurzbeschreibung": "Lorem ipsum",
        "meldung": "Lorem ipsum",
        "anhaenge": [
            "https://api-stage.lieferantcockpit.de/vsm/anhaenge/92A327D7-E632-4E94-AB63-2F1612EC0F7F",
            "https://api-stage.lieferantcockpit.de/vsm/anhaenge/6280393A-1550-4448-B5C5-6C277080042A",
            "https://api-stage.lieferantcockpit.de/vsm/anhaenge/F872A79A-FD9B-4444-B433-6254F404B08E"
        ]
    },
    {
        "lieferant": {
            "email": "max.mustermann@firma-xyz.de",
            "firmenname": "Firma XYZ GmbH",
            "lieferantcockpitId": "c10c2bd5-6011-4e50-9b7c-0665c6657aaa",
            "vorname": "Max",
            "nachname": "Mustermann",
            "telefonnummer": "0000-1234567"
        },
        "rahmenvertragid": "01042002-d16f-41e0-a7b1-3e8caa41a667",
        "erstelldatum": "2020-04-15T10:00:00.001Z",
        "kurzbeschreibung": "Lorem ipsum",
        "meldung": "Lorem ipsum",
        "anhaenge": [
            "https://api-stage.lieferantcockpit.de/vsm/anhaenge/2C79F2AE-5C76-464E-AEC0-9E49CAF4C77C"
        ]
    }
]
}

```

Codeblock 18: Response - JSON Endpunkt „VSM bereitstellen“

17.5. Attribut-Definitionen

Die im Datensatz ausgegebenen Werte beziehen sich auf die nachfolgenden Datenfelder. Diese Schnittstelle gibt lediglich Daten zurück. Demnach hat sie keine Pflichtfelder.

Attribut	Beschreibung	Typ
Lieferant		
email	E-Mail-Adresse des Lieferanten	String

firmenname	Name des Unternehmens	String
lieferantencockpitid	eindeutiges Identifikationsmerkmal des Lieferanten im Katalogsystem Diese GUID ist notwendig, um einen eindeutigen Bezug zum Lieferanten herstellen zu können.	GUID
vorname	Vorname des Lieferanten	String
nachname	Nachname des Lieferanten	String
telefonnummer	Telefonnummer des Lieferanten	String
Vertragsstörungsmeldung		
rahmenvertragid	eindeutiges Identifikationsmerkmal des Rahmenvertrags Diese GUID ist notwendig, um einen eindeutigen Bezug zum Rahmenvertrag herstellen zu können.	GUID
erstelldatum	Zeitpunkt der Erstellung der Vertragsstörungsmeldung	DateTime/String nach ISO_8601 Standard (z. B. 2020-01-16T14:30:00.001Z)
kurzbeschreibung	kurze Beschreibung der Vertragsstörungsmeldung	String (max. 40 Zeichen)
meldung	Beschreibung der Vertragsstörungsmeldung	String (max. 2000 Zeichen)
anhaenge	Array mit URLs zur Abholung der Anhänge Es enthält jeweils den vollständigen Pfad und eine GUID zur eindeutigen Identifizierung.	String-Array

Tabelle 42: Attribut-Definitionen der API „VSM bereitstellen“

18. Anlagen

18.1. Übergreifende Abbildungen

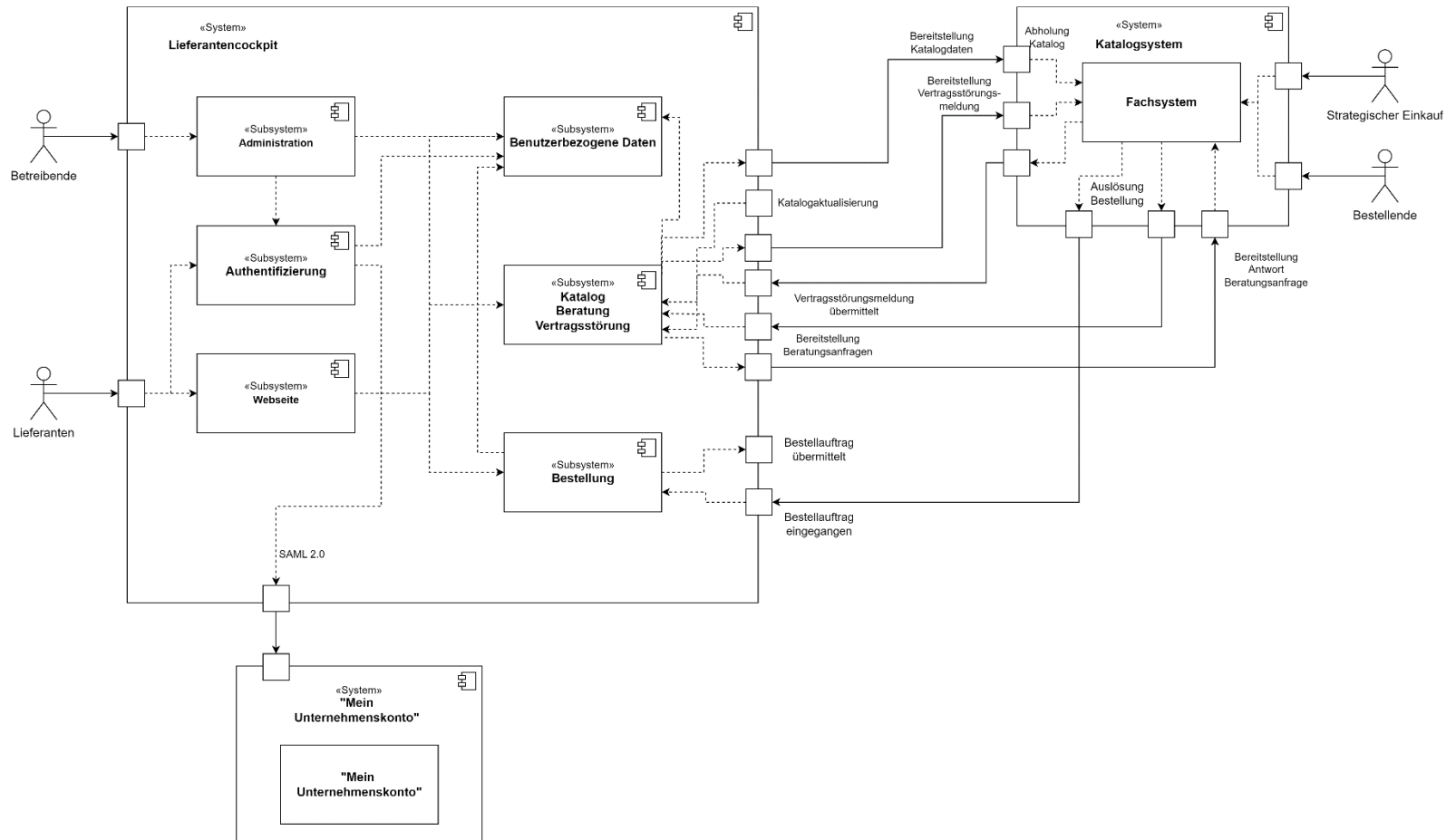


Abbildung 4: High Level Architektur des Lieferantencockpits (vergrößerte Darstellung)

18.2. Abbildungen Katalogmanagement

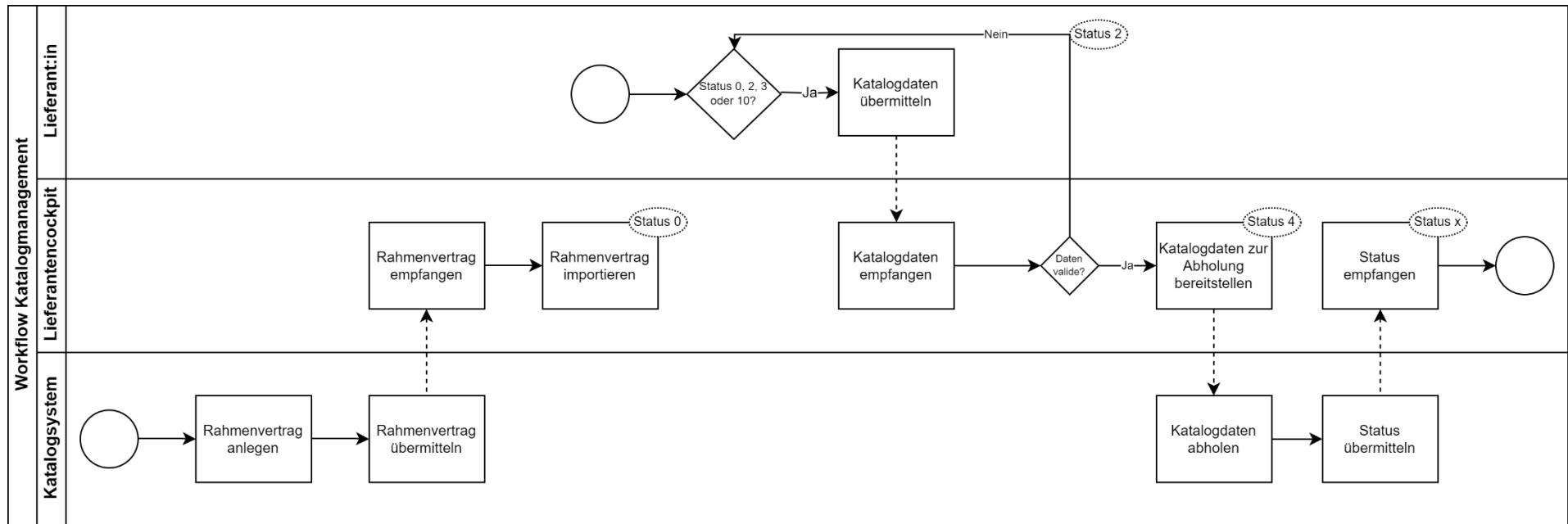


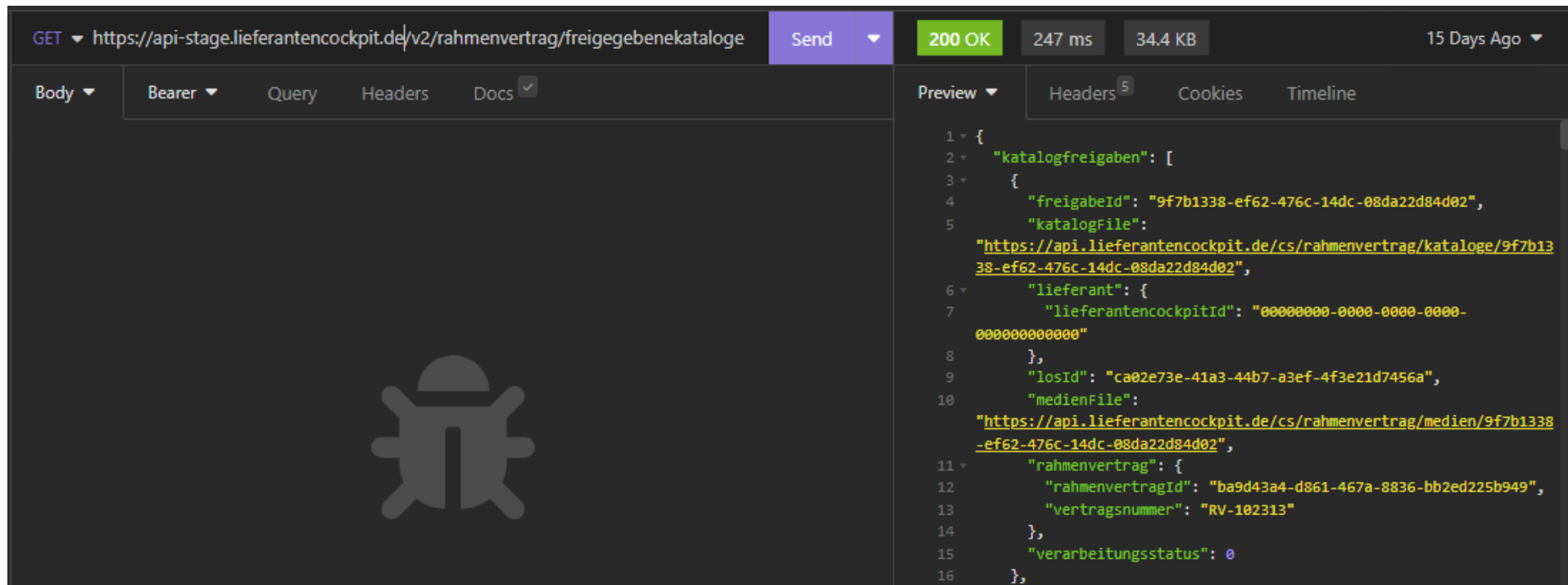
Abbildung 5: Workflow des Katalogmanagements (vergrößerte Darstellung)

```
POST https://api-stage.lieferantencockpit.de/v2/rahmenvertrag/empfangen Send
JSON Bearer Query Headers 2 Docs
1 {
2   "rahmenvertrag": {
3     "rahmenvertragId": "50e08cef-7f7c-4d35-9f39-37613b4a0a43",
4     "vertragsnummer": "RV Test API v2",
5     "vertragsbezeichnung": "Schreibtische und Bürostühle",
6     "vertragslaufzeitStart": "2023-01-01T00:00:00.000Z",
7     "vertragslaufzeitEnde": "2025-12-31T23:59:59.000Z",
8     "ansprechpersonName": "Rainer Glöckner",
9     "ansprechpersonEmail": "alexander.ruhland.ext@init.de",
10    "ansprechpersonTelefon": "0177 55431231"
11  },
12  "lieferanten": [
13    {
14      "lieferantencockpitId": ".lieferant.lieferantencockpitId",
15      "lose": [
16        {
17          "losId": "94bb0f60-02f5-4c62-af89-d2affcec3703"
18        }
19      ]
20    }
21  ],
22  "rahmenvertragspartner": {
23    "rahmenvertragspartnerId": "RVP-9400134",
24    "name": "Isabel Strickert",
25    "strasse": "Bergische Landstraße",
26    "hausnummer": "4",
27    "plz": "40629",
28    "stadt": "Düsseldorf"
29  }
30 }
```

Abbildung 6: Insomnia-Beispiel der API „Rahmenvertragsdaten empfangen“

```
PUT https://api-stage.lieferantcockpit.de/v2/rahmenvertrag/aktualisieren Send
JSON Bearer Query Headers 2 Docs
1 {
2   "rahmenvertrag": {
3     "rahmenvertragId": "4708f224-6f97-443a-9a09-e56f1e09e89b",
4     "vertragsnummer": "RV Test API v2",
5     "vertragsbezeichnung": "Schreibtische und Bürostühle",
6     "vertragslaufzeitStart": "2023-01-01T00:00:00.000Z",
7     "vertragslaufzeitEnde": "2025-12-31T23:59:59.000Z",
8     "ansprechpersonName": "Rainer Glöckner",
9     "ansprechpersonEmail": "alexander.ruhland.ext@init.de",
10    "ansprechpersonTelefon": "0177 55431231"
11  },
12  "lieferanten": [
13    {
14      "lieferantcockpitId": ".lieferant.lieferantcockpitId ",
15      "lose": [
16        {
17          "losId": "74aa10bb-31d2-4944-be8c-635970f56383"
18        }
19      ]
20    }
21  ],
22  "rahmenvertragspartner": {
23    "rahmenvertragspartnerId": "RVP-9400134",
24    "name": "Isabel Strickert",
25    "strasse": "Bergische Landstraße",
26    "hausnummer": "4",
27    "plz": "40629",
28    "stadt": "Düsseldorf"
29  }
30 }
```

Abbildung 7: Insomnia-Beispiel der API „Rahmenvertragsdaten aktualisieren“



GET <https://api-stage.lieferantcockpit.de/v2/rahmenvertrag/freigegebenekataloge> Send 200 OK 247 ms 34.4 KB 15 Days Ago

Body Bearer Query Headers Docs

Preview Headers Cookies Timeline

```
1 {
2   "katalogfreigaben": [
3     {
4       "freigabeId": "9f7b1338-ef62-476c-14dc-08da22d84d02",
5       "katalogFile":
6         "https://api.lieferantcockpit.de/cs/rahmenvertrag/kataloge/9f7b1338-ef62-476c-14dc-08da22d84d02",
7       "lieferant": {
8         "lieferantcockpitId": "00000000-0000-0000-0000-000000000000"
9       },
10      "losId": "ca02e73e-41a3-44b7-a3ef-4f3e21d7456a",
11      "medienFile":
12        "https://api.lieferantcockpit.de/cs/rahmenvertrag/medien/9f7b1338-ef62-476c-14dc-08da22d84d02",
13      "rahmenvertrag": {
14        "rahmenvertragId": "ba9d43a4-d861-467a-8836-bb2ed225b949",
15        "vertragsnummer": "RV-102313"
16      },
17      "verarbeitungsstatus": 0
18    }
19  ]
20 }
```

Abbildung 8: Insomnia-Beispiel der API „Katalogbereitstellung“

The screenshot shows the Insomnia REST client interface. The top bar indicates the project name 'Lieferantencockpit' and the current request method 'POST' to the endpoint '...APIGATEWAY_URL /cs/Rahmenvertrag/Statusanpassung'. The response status is '200 OK' with a response time of '431 ms' and '0 B' of data.

The request body is shown in the JSON tab:

```

1 - {
2 -   "katalogSystem": {
3 -     "rahmenvertrag": {
4 -       "rahmenvertragId": "b0bd9994-2abb-417f-9a1f-e223fcccb556",
5 -       "vertragsnummer": "RV-71234567"
6 -     }
7 -   },
8 -   "katalogfreigabe": {
9 -     "freigabeId": null,
10 -    "losId": "4a297f94-b46b-462f-be3c-413a26b220d7",
11 -    "status": "...Rahmenvertrag.KatalogfreigabeStatus.Katalogfreigabe"
12 -  }
13 - }

```

The response body is shown in the JSON tab:

```

{
  "katalogSystem": {
    "rahmenvertrag": {
      "rahmenvertragId": "b0bd9994-2abb-417f-9a1f-e223fcccb556",
      "vertragsnummer": "RV-71234567"
    }
  },
  "katalogfreigabe": {
    "freigabeId": null,
    "losId": "4a297f94-b46b-462f-be3c-413a26b220d7",
    "status": 8
  }
}

```

The right-hand side of the interface shows the response details, including the status '200 OK', headers, and cookies. The response body is displayed as a JSON object.

Abbildung 9: Insomnia-Beispiel der API „Statusanpassungen empfangen“

18.3. Abbildungen Beratung

```
POST https://api-stage.lieferantcockpit.de/v1/beratung/erstellen Send 200 OK 4.08 s 0 B
JSON Bearer Query Headers 1 Docs Headers 4
1 {
2   "beratungsanfrage": {
3     "beratungsanfrageId": "D7GA4KL87SI",
4     "datum": "2023-02-09T11:55:01.000Z",
5     "dienststelle": "Dienststelle ABC Live-Demo 3",
6     "bestellerVorname": "Jochen",
7     "bestellerNachname": "Rumler",
8     "bestellerEmail": "alexander.ruhland.ext@init.de",
9     "kurzbezeichnung": "Frage zu Bürostuhl aus Katalogsystem",
10    "text": "Gibt es den Bürostuhl in gelb auch mit Lehne?"
11  },
12  "lieferant": {
13    "lieferantcockpitId": "...lieferant.lieferantcockpitId",
14    "email": "...lieferant.email"
15  },
16  "warenkorb": [
17    {
18      "position": 1,
19      "artikelbezeichnung": "Bürostuhl der Firma XY",
20      "identifikatorLieferant": "1200120012",
21      "menge": "4",
22      "merkmale": [
23        {
24          "eigenschaft": "Farbe",
25          "wert": "Schwarz"
26        },
27        {
28          "eigenschaft": "Material",
29          "wert": "Netzstoff"
30        },
31        {
32          "eigenschaft": "Höhe in cm",
33          "wert": "120,0"
34        },
35        {
36          "eigenschaft": "Belastbarkeit in kg",
37          "wert": "120"
38        }
39      ]
40    }
41  ]
42 }
```

No body returned for response

Abbildung 10: Insomnia-Beispiel der API „Beratungsanfragen empfangen“

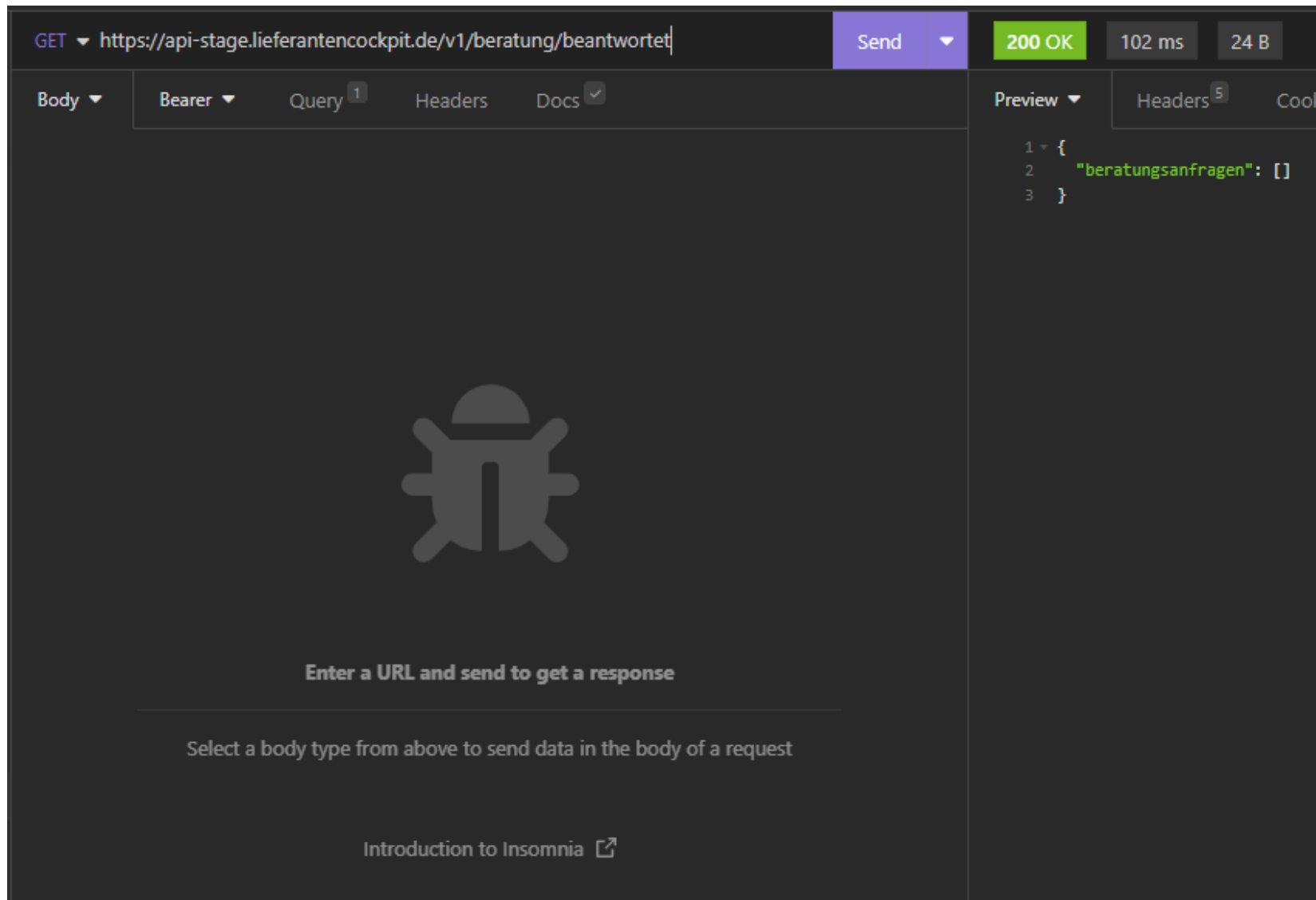


Abbildung 11: Insomnia-Beispiel der API „Beratungsanfragen beantwortet“

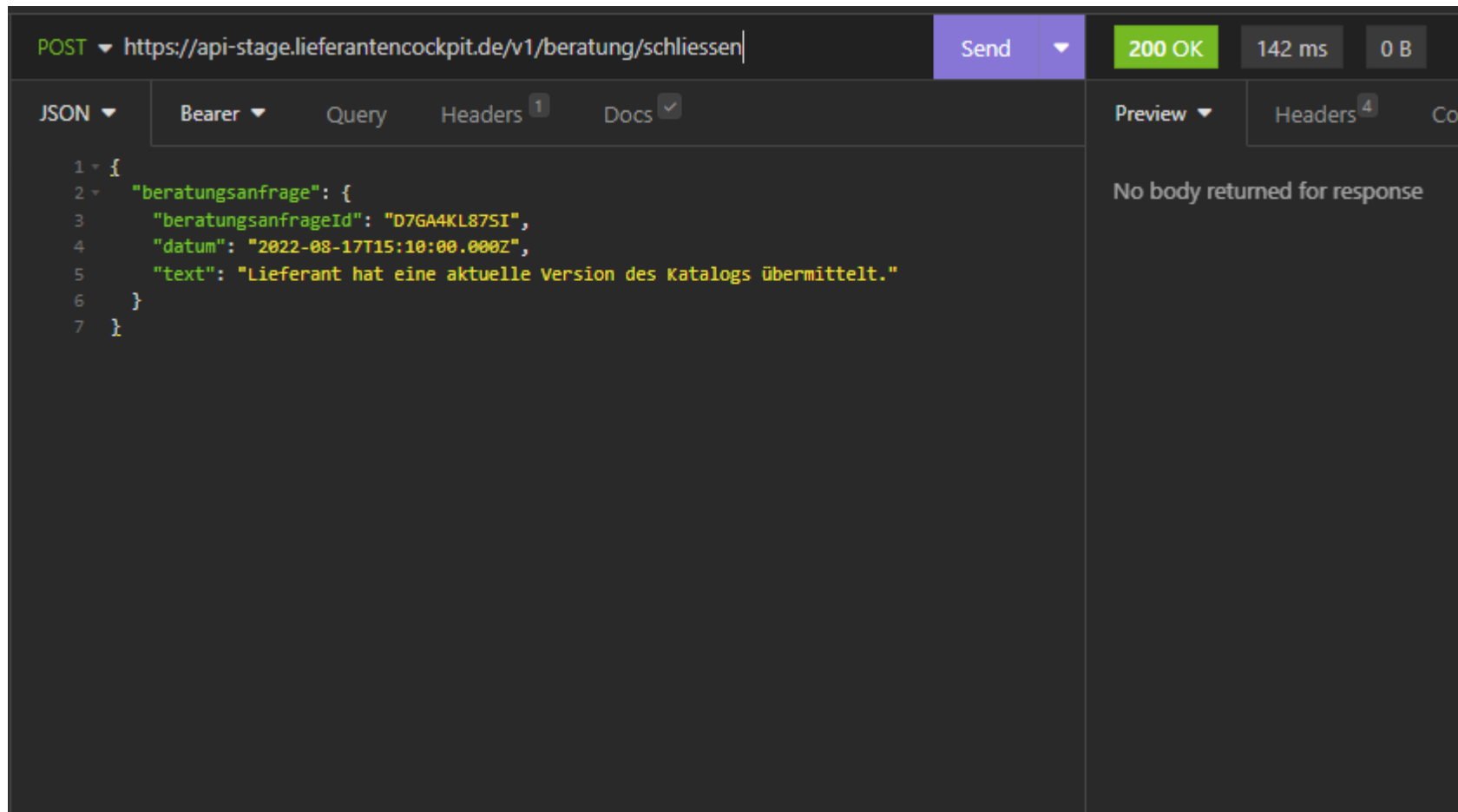


Abbildung 12: Insomnia-Beispiel der API „Geschlossene Beratungsanfragen empfangen“

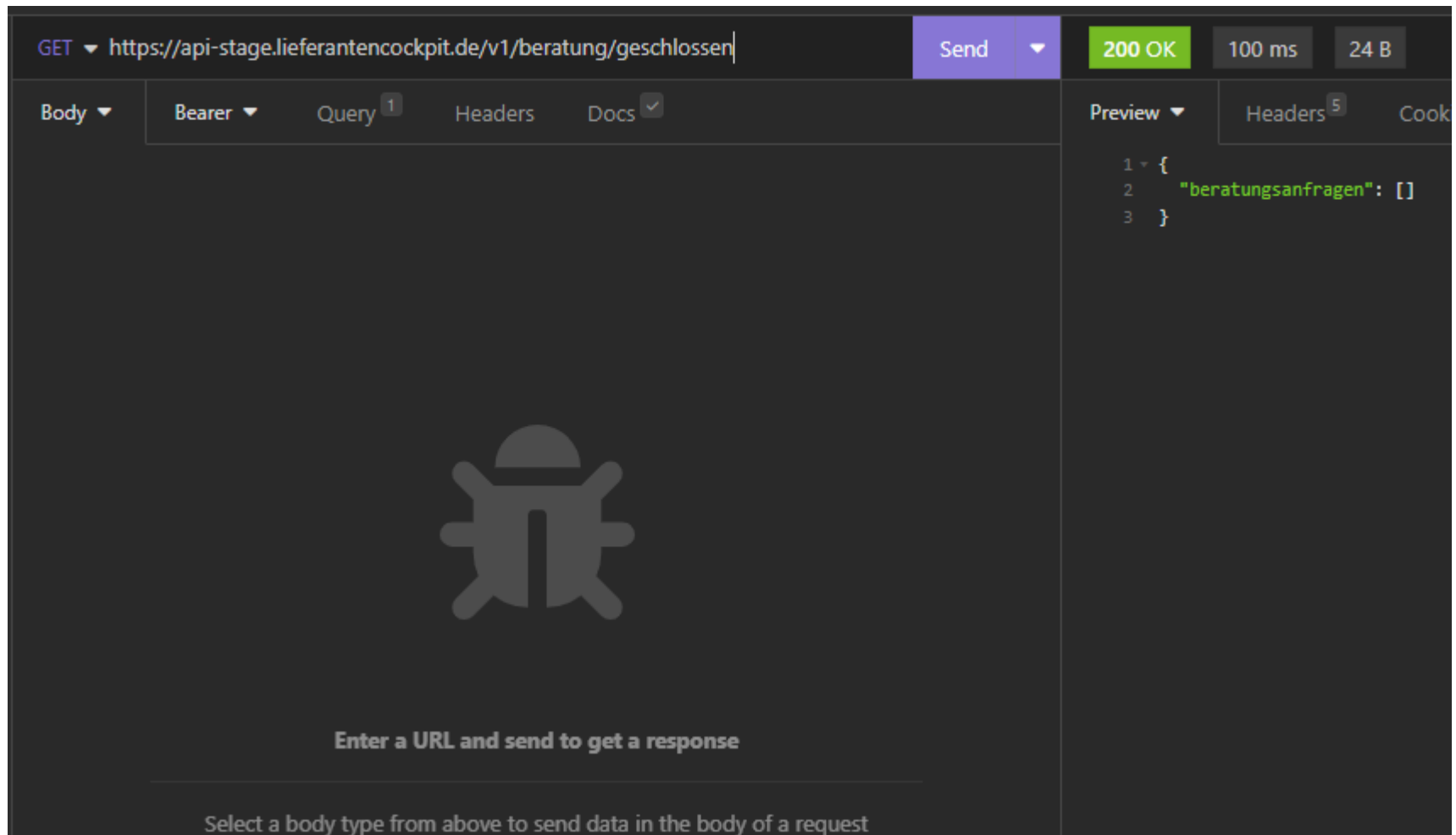


Abbildung 13: Insomnia-Beispiel der API „Geschlossene Beratungsanfragen bereitstellen“

18.4. Abbildungen Bestellung

The screenshot displays the Insomnia API client interface. The top bar shows the project name 'Lieferantencockpit' and the current request details: 'GET /bestellung/bereitstellung' with a status of '200 OK', a response time of '3.73 s', and a size of '15.9 KB'. The left sidebar contains a navigation menu with items like 'Administration', 'Lieferantencockpit', 'User Related Data Service', 'API Gateway', 'Authentication Service', 'Catalogue Service', 'Consultation Service', and 'Purchase Order Service'. The main area is divided into 'Body' and 'Preview' sections. The 'Body' section shows the request body with a large bug icon and the text 'Enter a URL and send to get a response'. The 'Preview' section shows the response body, which is a JSON object containing a long string of characters. The response headers are also visible, including 'Date: Fri, 18 Nov 2022 08:53:35 GMT', 'Server: Apache', 'content-length: 16266', 'content-type: application/json; charset=utf-8', and 'Via: 1.1 api-stage.lieferantencockpit.de'.

Abbildung 14: Insomnia-Beispiel der API „Bestellung bereitstellen“

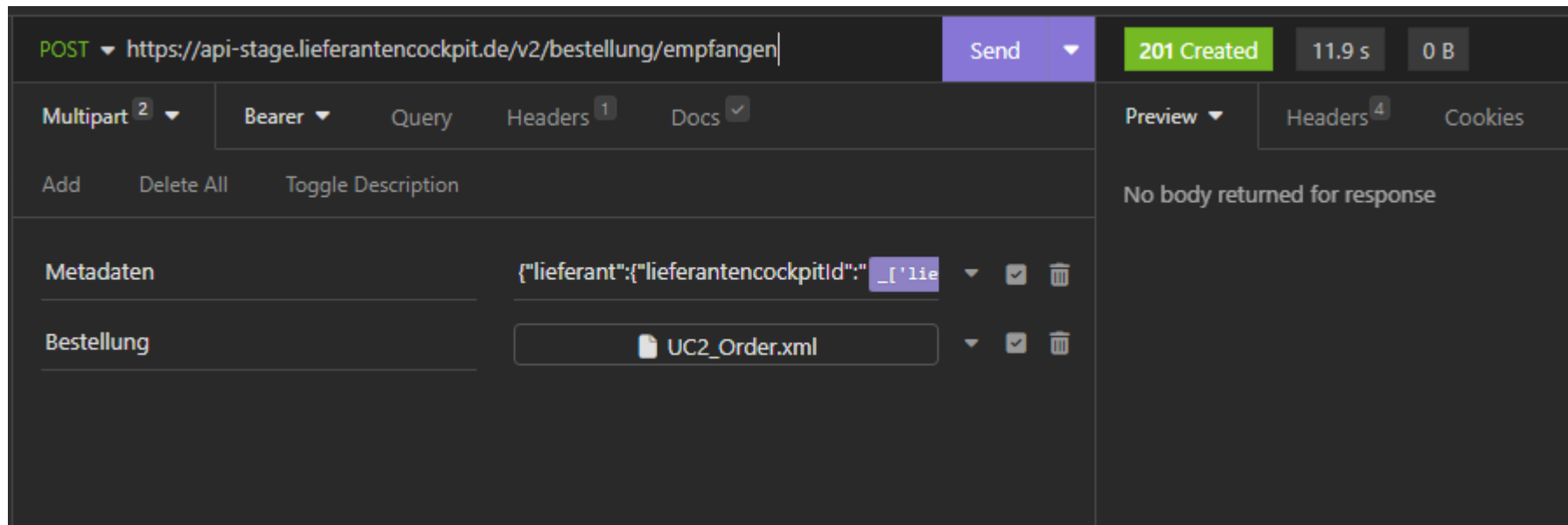


Abbildung 15: Insomnia-Beispiel der API „Bestellung empfangen“

