



[www.datenlink.info](http://www.datenlink.info)

---

# datenlink-Schnittstelle

## Version 1.03

## Inhalt

<b>1 Allgemeines</b>	<b>3</b>
1.1 Datenaustausch.....	3
1.2 Zugriffstypen.....	3
<b>2 Format der Rückgabewerte</b>	<b>4</b>
2.1 HTTP-Statuscodes .....	4
2.2 Rückgabewerte .....	5
2.3 Beispielryckgaben für unterschiedliche Statuscodes .....	5
2.3.1 Erfolgreiche Serviceabfrage (HTTP-Statuscode 200).....	5
2.3.2 Fehlerhafte Anfrage (HTTP-Statuscode 400).....	5
2.3.3 Fehlerhafte API-Daten (HTTP-Statuscode 403) .....	6
2.3.4 Service nicht vorhanden (HTTP-Statuscode 501).....	6
2.3.5 Service temporär nicht verfügbar (HTTP-Statuscode 503).....	6
<b>3 Programm-Services</b>	<b>7</b>
3.1 DatenlinkConnector::authenticateUser .....	7
3.2 DatenlinkConnector::getCompanyAPIToken .....	7
3.3 DatenlinkConnector::suggestSIDIngredient .....	8
3.4 DatenlinkConnector:: getComponentList .....	8
3.4.1 DatenlinkConnector::getLanguagesList .....	8
3.4.2 DatenlinkConnector::getAdditivesList .....	8
3.4.3 DatenlinkConnector::getOriginOilsList .....	9
3.4.4 DatenlinkConnector::getAllergenesList .....	9
3.4.5 DatenlinkConnector::getOriginCountriesList .....	9
3.5 DatenlinkConnector::getSingleComponent .....	10
3.5.1 DatenlinkConnector::getAdditiveById.....	10
3.5.2 DatenlinkConnector::getAdditiveByName .....	10
3.5.3 DatenlinkConnector::getOriginOilById .....	11
3.5.4 DatenlinkConnector::getOriginOilByName .....	11
3.5.5 DatenlinkConnector::getAllergenById.....	11
3.5.6 DatenlinkConnector::getAllergenByName .....	12
3.5.7 DatenlinkConnector::getIngredientById .....	12

---

3.5.8 DatenlinkConnector::getIngredientByName .....	13
3.6 DatenlinkConnector:: getCompanyData .....	14
<b>4 Unternehmens-Services</b>	<b>15</b>
4.1 DatenlinkConnector::getActiveSubscribedProducts.....	15
4.2 DatenlinkConnector::getProductData .....	16

## 1 Allgemeines

Die offizielle datenlink-Schnittstelle stellt verschiedene Funktionen bereit, die es Unternehmen ermöglichen, mittels externer Programme auf Ihre bei datenlink hinterlegten Daten zuzugreifen.

### 1.1 Datenaustausch

Die Kommunikation zwischen Client (externem Programm) und datenlink-Server erfolgt über HTTP. Der Client sendet seine JSON-formatierten Anfragen via POST-Methode an den Server und erhält die Ergebnisse ebenfalls im JSON-Format. Für die Skriptsprache PHP stellt datenlink bereits eine vorgefertigte Klasse bereit (beachten Sie bitte auch die Kommentare innerhalb der Datei), die Sie individuell nach Ihren Ansprüchen anpassen können. Für andere Programmiersprachen müssen Sie ggf. weitere Anpassungen vornehmen.

### 1.2 Zugriffstypen

Es existieren zwei verschiedene Zugriffsebenen auf die Schnittstelle, die sich in der Bereitstellung der aufrufbaren Services unterscheiden.

Der erste Zugriffstyp beinhaltet die Programm-Services und wird verwendet, wenn Sie ein externes Programm anbieten, welches multimandantenfähig ist und Sie die Benutzer anhand derer datenlink-Logindaten authentifizieren müssen, um Zugriff auf den Company-API-Token (CAT) zu bekommen. Voraussetzung zur Nutzung dieser Funktionalität ist ein gültiger Program-API-Token (PAT), den Sie unter Angabe Ihrer Verwendung unter [info@datenlink.info](mailto:info@datenlink.info) beantragen können.

Der zweite Zugriffstyp ermöglicht den Zugriff auf die Unternehmens-Services. Voraussetzung dafür ist, dass Sie einen gültigen CAT besitzen, den Sie bei jeder Verbindungskennung angeben müssen. Mit einem gültigen CAT können Sie auf die Unternehmens-Services des dem CAT zugehörigen Unternehmens zugreifen.

Die Abschnitte 3 und 4 beschreiben die jeweiligen Services mit deren Eingabeparameter und Rückgabewerten.

## 2 Format der Rückgabewerte

Die Rückgabewerte aller Anfragen an den Server werden JSON-formatiert ausgegeben. Der Content-Type für alle Rückgabewerte lautet *application/json*.

### 2.1 HTTP-Statuscodes

Mittels der HTTP-Statuscodes können Sie feststellen, ob Ihre Anfrage an den datenlink-Server erfolgreich war und ob Ihr angefragter Service erfolgreich aufgerufen wurde. Die datenlink-Schnittstelle liefert Ihnen folgende Statuscodes zurück:

Statuscode	Statustext	Bedeutung
200	OK	Ihre Anfrage wurde erfolgreich ausgeführt und der Service wurde erfolgreich aufgerufen. Nun müssen Sie noch die Rückgabewerte Ihres aufgerufenen Services auswerten. Details dazu entnehmen Sie bitte dem Abschnitt des jeweiligen Services.
400	Bad Request	Sie haben eine ungültige Anfrage gesendet. Der Rückgabewert des Servers verrät Ihnen, welche Parameter für Ihre Anfrage gesetzt werden müssen.
403	Forbidden	Sie haben einen ungültigen API-Token übermittelt und erhalten deshalb keinen Zugriff auf die Schnittstelle.
501	Not Implemented	Sie haben einen Service angefordert, der generell oder speziell für Ihre Zugriffsrechte nicht verfügbar ist.
503	Service Unavailable	Der Service ist temporär nicht verfügbar. Versuchen Sie bitte einen erneuten Aufruf des Services nach der im Antwort-Header unter <i>Retry-After</i> angegebenen Zeit (in Sekunden).

## 2.2 Rückgabewerte

Neben den HTTP-Statuscodes spielen auch die Rückgabewerte eine wichtige Rolle zum Überprüfen, ob eine Anfrage erfolgreich war. Jede Abfrage liefert den Array-Wert *status* zurück. Dieser kann entweder den Wert „success“, bei Erfolg, oder den Wert „error“, bei einem aufgetretenen Fehler, annehmen. Die Werte unterhalb von *data* sind servicespezifisch. Welche Rückgaben es geben kann, entnehmen Sie bitte dem jeweilig dem Service zugeordneten Abschnitt. Im Fehlerfall enthält *data* die Fehlerbeschreibung.

## 2.3 Beispielrückgaben für unterschiedliche Statuscodes

In diesem Abschnitt finden Sie einige Beispielrückgaben der Schnittstelle.

### 2.3.1 Erfolgreiche Serviceabfrage (HTTP-Statuscode 200)

Dieses Ergebnis zeigt die erfolgreiche Abfrage der Benutzerauthentifizierung durch die Methode *DatenlinkConnector::authenticateUser*

HTTP-Statuscode: 200

```
Array
(
    [status] => success
    [data] =>
        Array
        (
            [auth_status] => 0
            [hint] => authentication successful
            [company_id] => 5
            [user_id] => 5
        )
)
```

### 2.3.2 Fehlerhafte Anfrage (HTTP-Statuscode 400)

Dieses Ergebnis zeigt die Rückgabe einer fehlerhaft gestellten Anfrage.

HTTP-Statuscode: 400

```
Array
(
    [status] => error
    [data] => not all required data-parameters are set. require the
following data-parameter: auth_token, service, data
)
```

### 2.3.3 Fehlerhafte API-Daten (HTTP-Statuscode 403)

Dieses Ergebnis zeigt die Rückgabe bei fehlerhaften API-Logindaten.

HTTP-Statuscode: 403

```
Array
(
    [status] => error
    [data] => you are not allowed to access this content
)
```

### 2.3.4 Service nicht vorhanden (HTTP-Statuscode 501)

Dieses Ergebnis zeigt die Rückgabe des Aufrufs eines nicht implementierten Services.

HTTP-Statuscode: 501

```
Array
(
    [status] => error
    [data] => the requested service is not implemented
)
```

### 2.3.5 Service temporär nicht verfügbar (HTTP-Statuscode 503)

Dieses Ergebnis zeigt die Rückgabe einer Anfrage, bei der seitens datenlink ein interner Fehler aufgetreten ist.

HTTP-Statuscode: 503

```
Array
(
    [status] => error
    [data] => your requested service is temporary not available. please
    retry after 86400 seconds
)
```

### 3 Programm-Services

Nachfolgend sind die einzelnen Programm-Services aufgelistet, die Ihnen mit einem gültigen PAT zur Verfügung stehen.

**Beachten Sie bitte, dass Sie den PAT bei jeder Anfrage zur Authentifizierung unter dem Parameter-Namen *auth\_token* mitsenden müssen.**

#### 3.1 DatenlinkConnector::authenticateUser

Diese Methode ermöglicht es Ihnen, die Gültigkeit der Logindaten von datenlink-Benutzern zu verifizieren.

##### Request-Parameter:

- *user* – enthält die Email-Adresse des Benutzers (String)
- *password* – enthält das Passwort des Benutzers (String)

##### Rückgabewerte:

- *auth\_status* – gibt an, ob ein Benutzer erfolgreich authentifiziert wurde:
  - 0 = Authentifizierung erfolgreich
  - 1 = Benutzerdaten korrekt, jedoch ist der Benutzer gesperrt
  - 2 = Benutzername ist richtig, Passwort ist ungültig
  - 3 = Benutzername ist nicht bei datenlink vorhanden
- *hint* – Erläuterung des *auth\_status* in Textform (String)
- *company\_id* – zugehörige Unternehmens-ID des Benutzers (Integer)
- *user\_id* – Benutzer-ID (Integer)

#### 3.2 DatenlinkConnector::getCompanyAPIToken

Diese Methode ermöglicht es Ihnen, den CAT für valide Benutzer-Logindaten auszu-lesen.

Die Request- und Rückgabewerte sind identisch zu denen der Methode *Datenlink-Connector::authenticateUser*, jedoch besitzt diese Methode noch den zusätzlichen Rückgabewert

- *api\_token* – CAT. Besitzt der Unternehmen keinen CAT, wird -1 zurückge-geben. (String/Integer)



### 3.3 DatenlinkConnector::suggestSIDIngredient

Dieser Service ermöglicht es Ihnen, einen Rohstoff für die SID vorzuschlagen.

#### Request-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)
- *ingredient* (String)
- *additional* (String)

**Rückgabewert: 1 (Erfolg) oder 0 (Fehler)**

### 3.4 DatenlinkConnector::getComponentList

Dieser Service stellt den Grundservice für die nachfolgend aufgelisteten Methoden dar. Sie können ihn direkt aufrufen, indem Sie alle nötigen Parameter setzen, oder eine der nachfolgenden Methoden aufrufen.

#### Request-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)
- *type* (String)

**Rückgabewerte: (assoziatives Array, bestehend aus allen Komponenten)**

- Key: SID-ID (Integer/String)
- Wert: Name der Komponente

#### 3.4.1 DatenlinkConnector::getLanguagesList

Ermöglicht das Abfragen der verfügbaren Sprachen der SID in der übermittelten Sprache.

#### Methoden-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)

**Rückgabewerte: (assoziatives Array, bestehend aus allen Komponenten)**

- Key: Sprachcode nach ISO 639-3 (String)
- Wert: Name der Sprache

#### 3.4.2 DatenlinkConnector::getAdditivesList

Ermöglicht das Abfragen der verfügbaren Zusatzstoffe der SID in der übermittelten Sprache.

#### Methoden-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)

**Rückgabewerte:**

- Key: SID-ID (Integer)
- Wert: Name des Zusatzstoffes

**3.4.3 DatenlinkConnector::getOriginOilsList**

Ermöglicht das Abfragen der verfügbaren Ursprungsöle der SID in der übermittelten Sprache.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)

**Rückgabewerte:**

- Key: SID-ID (Integer)
- Wert: Name des Ursprungsöls

**3.4.4 DatenlinkConnector::getAllergenesList**

Ermöglicht das Abfragen der verfügbaren Allergene der SID in der übermittelten Sprache.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)

**Rückgabewerte:**

- Key: SID-ID (Integer)
- Wert: Name des Allergenes

**3.4.5 DatenlinkConnector::getOriginCountriesList**

Ermöglicht das Abfragen der verfügbaren Ursprungsländer der SID in der übermittelten Sprache.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)

**Rückgabewerte:**

- Key: Ländercode (String, Ländercode nach ISO 3166-1)
- Wert: Name des Allergenes

### 3.5 DatenlinkConnector::getSingleComponent

Dieser Service stellt den Grundservice für die nachfolgend aufgelisteten Methoden dar. Sie können ihn direkt aufrufen, indem Sie alle nötigen Parameter setzen, oder eine der nachfolgenden Methoden aufrufen.

#### Request-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)
- *type* (String – Art der Komponente)
- *order\_format* (String, „id“ oder „name“)
- *value* (Integer/String – Prüfwert)
- *additional\_value* (Integer – zusätzlicher Prüfwert)

#### Rückgabewerte: (Array der folgenden Form, mit Ausnahme von 3.4.7 / 3.4.8)

- Key: ID (Integer)
- Wert: Name des Stoffes (String)

#### 3.5.1 DatenlinkConnector::getAdditiveById

Mit diesem Service kann ein Zusatzstoff anhand einer gegebenen ID ermittelt werden.

#### Methoden-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)
- *id* (Integer – ID des Zusatzstoffes)

#### Rückgabewerte:

- Key: Zusatzstoff-ID (Integer)
- Wert: Name des Zusatzstoffes (String)

#### 3.5.2 DatenlinkConnector::getAdditiveByName

Mit diesem Service kann ein Zusatzstoff anhand eines gegebenen Namens ermittelt werden. Die Suche ist unabhängig von Groß- und Kleinschreibung, die Schreibweise muss jedoch exakt sein. Der Eingabename muss in der in *lang* übermittelten Sprache angegeben werden. Die Rückgabe erfolgt ebenfalls in dieser Sprache.

#### Methoden-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)
- *value* (String – Name des Zusatzstoffes)

**Rückgabewerte:**

- Key: Zusatzstoff-ID (Integer)
- Wert: Name des Zusatzstoffes (String)

**3.5.3 DatenlinkConnector::getOriginOilById**

Mit diesem Service kann ein Ursprungsöl anhand einer gegebenen ID ermittelt werden.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)
- *id* (Integer – ID des Ursprungsöls)

**Rückgabewerte:**

- Key: Ursprungsöl-ID (Integer)
- Wert:
  - Key 1: *name*
  - Wert 1: Name des Ursprungsöls (String)

**3.5.4 DatenlinkConnector::getOriginOilByName**

Mit diesem Service kann ein Ursprungsöl anhand eines gegebenen Namens ermittelt werden. Die Suche ist unabhängig von Groß- und Kleinschreibung, die Schreibweise muss jedoch exakt sein. Der Eingabename muss in der in *lang* übermittelten Sprache angegeben werden. Die Rückgabe erfolgt ebenfalls in dieser Sprache.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)
- *value* (String – Name des Ursprungsöls)

**Rückgabewerte:**

- Key: Ursprungsöl-ID (Integer)
- Wert:
  - Key 1: *name*
  - Wert 1: Name des Ursprungsöls (String)

**3.5.5 DatenlinkConnector::getAllergenById**

Mit diesem Service kann ein Allergen anhand einer gegebenen ID ermittelt werden.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)
- *id* (Integer – ID des Allergenes)

**Rückgabewerte:**

- Key: Allergen-ID (Integer)
- Wert:
  - Key 1: *name*
  - Wert 1: Name des Allergenes (String)

**3.5.6 DatenlinkConnector::getAllergenByName**

Mit diesem Service kann ein Allergen anhand eines gegebenen Namens ermittelt werden. Die Suche ist unabhängig von Groß- und Kleinschreibung, die Schreibweise muss jedoch exakt sein. Der Eingabename muss in der in *lang* übermittelten Sprache angegeben werden. Die Rückgabe erfolgt ebenfalls in dieser Sprache.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)
- *value* (String – Name des Allergenes)

**Rückgabewerte:**

- Key: Allergen-ID (Integer)
- Wert:
  - Key 1: *name*
  - Wert 1: Name des Allergenes (String)

**3.5.7 DatenlinkConnector::getIngredientById**

Mit diesem Service kann ein Rohstoff anhand einer gegebenen ID ermittelt werden.

**Methoden-Parameter:**

- *lang* (String, Sprachcode nach ISO 639-3)
- *id* (Integer – ID des Allergenes)
- *name\_id* (Integer – Namensklassifizierungs-ID des Rohstoffes, optional)

**Rückgabewerte:**

- Key: Allergen-ID (Integer)
- Wert:
  - Key 1: *name*

- Wert 1: Name des Rohstoffes
- Key 2: *name\_id*
- Wert 2: Namensklassifizierungs-ID des Rohstoffes
- Key 3: *is\_additive*
- Wert 3: Angabe, ob Rohstoff ein Zusatzstoff ist (1 = ja, 0 = nein)
- Key 4: *has\_oilorigin*
- Wert 4: Angabe, ob Rohstoff pflanzl. Öl als Ursprung besitzt (1 = ja, 0 = nein)

### 3.5.8 DatenlinkConnector::getIngredientByName

Mit diesem Service kann ein Rohstoff anhand eines gegebenen Namens ermittelt werden. Die Suche ist unabhängig von Groß- und Kleinschreibung, die Schreibweise muss jedoch exakt sein. Der Eingabename muss in der in *lang* übermittelten Sprache angegeben werden. Die Rückgabe erfolgt ebenfalls in dieser Sprache.

#### Methoden-Parameter:

- *lang* (String, Sprachcode nach ISO 639-3)
- *value* (String – Name der Zutat)

#### Rückgabewerte:

- Key: Allergen-ID (Integer)
- Wert:
  - Key 1: *name*
  - Wert 1: Name des Rohstoffes
  - Key 2: *name\_id*
  - Wert 2: Namensklassifizierungs-ID des Rohstoffes
  - Key 3: *is\_additive*
  - Wert 3: Angabe, ob Rohstoff ein Zusatzstoff ist (1 = ja, 0 = nein)
  - Key 4: *has\_oilorigin*
  - Wert 4: Angabe, ob Rohstoff pflanzl. Öl als Ursprung besitzt (1 = ja, 0 = nein)

### 3.6 DatenlinkConnector:: getCompanyData

Diese Methode ermöglicht die Zuordnung der in den datenlink-Dateien enthaltenen Unternehmens-IDs (*foodxml\_company*-Tag) zum Namen und der Anschrift des Datengebers.

#### Request-Parameter:

- *company\_id* (Integer)

#### Rückgabewerte:

- *company* – Name des Unternehmens (String)
- *street* – Straßename (String)
- *streetnumber* – Hausnummer (String)
- *zipcode* – Postleitzahl (String)
- *city* – Stadt (String)
- *country* – Land (String)

## 4 Unternehmens-Services

Nachfolgend sind die einzelnen Unternehmens-Services aufgelistet, die Ihnen mit einem gültigen CAT zur Verfügung stehen. Sollten Sie noch keinen CAT für Ihr Unternehmen angelegt haben, können Sie diesen unter *Einstellungen* → *Company-API-Token verwalten* erstellen.

**Beachten Sie bitte, dass Sie den CAT bei jeder Anfrage zur Authentifizierung unter dem Parameter-Namen *auth\_token* mitsenden müssen.**

### 4.1 DatenlinkConnector::getActiveSubscribedProducts

Diese Methode liefert Ihnen die aktuell verarbeiteten Versionen der im Reader abonnierten Produkte des dem CAT zugeordneten Unternehmens zurück. Die Reihenfolge der Ausgabe ist nach dem Produktnamen aufsteigend sortiert.

**Request-Parameter: -**

**Rückgabewerte: (Array, bestehend aus N<sup>1</sup> Produkten)**

- *productname* – Name des Produkts (String)
- *labeling\_method* – Produktkennung: Methodik der Zuordnung zum Gebinde (String)
- *labeling\_value* – Produktkennung: Kennzeichnung (String)
- *tradenname* – Verkehrsbezeichnung (String)
- *productnumber* – Artikelnummer (String)
- *company* – Herstellername (String)
- *company\_id* – Hersteller-ID (Integer)
- *datenlink\_id* – datenlink-ID (String)
- *xmlfile* – URL zur datenlink-Datei (String)
- *pdffile* – URL zur zusätzlich hinterlegten Produktspezifikation. Ist keine Spezifikation vorhanden, ist der Wert -1 enthalten. (String/Integer)

---

<sup>1</sup> Anzahl der im Reader abonnierten Produkte



## 4.2 DatenlinkConnector::getProductData

Diese Methode liefert Ihnen weitere produktspezifische Daten für das Produkt mit der von Ihnen übergebenen datenlink-ID zurück.

### Request-Parameter:

- *datenlink\_id* (String)

### Rückgabewerte:

- *datenlink\_id* – datenlink-ID (String)
- *software* – Erzeugersoftware der datenlink-Datei (String)
- *xmlfile* – URL zur datenlink-Datei (String)
- *base64\_xmlstring* – Der Inhalt der datenlink-Datei als base64-codierter String (String)
- *pdffile* – URL zur zusätzlich hinterlegten Produktspezifikation. Ist keine Spezifikation vorhanden, ist der Wert -1 enthalten. (String/Integer)