



# Deutscher Elektronischer Sequenzdaten-Hub

Spezifikation der Bereitstellung über REST API

Version 1.0

27.04.2021

# 1 Einleitung

Im Rahmen des [Deutscher Elektronischer Sequenzdaten-Hub \(DESH\)](#) werden neben der Übermittlung rekonstruierten Sequenzdaten via [Web-Oberfläche](#) auch eine Übermittlung via REST API Schnittstellen bereitgestellt.

In diesem Dokument werden die beiden geplanten Schritte zur Übermittlung der Daten beschrieben. Eine explizite Dokumentation der Schnittstelle erfolgt über die mitgelieferte Datei „receiver.yaml“ und die Nutzung über <https://editor.swagger.io>.

## 2 Schritte zur Übermittlung der Daten

### 2.1 Voraussetzungen zur Nutzung

Für die Bereitstellung der Daten über eine REST API gelten die gleichen Voraussetzungen wie für die Nutzung der Web-Oberfläche. Es werden zwei Faktoren authentifiziert: ein gültiges TLS-Zertifikat sowie eine Benutzername/Passwort-Kombination (Account). Mit Hilfe des Accounts (der Benutzername/Passwort-Kombination) kann ein Access-Token angefordert werden, welcher zur Authentifizierung gegenüber der DESH-Anwendung notwendig ist.

Um per API einen Token zu erhalten, muss der in der mitgelieferten „receiver.yaml“ unter dem Punkt „Authorization“ aufgeführte Request genutzt werden. Folgende Parameter müssen mit gesendet werden:

Parameter	Testumgebung	Produktivumgebung
client_id	desh-api	desh-api
client_secret	8245c565-535f-4ded-aeb1-20e4c8c2ed32	(wird Ihnen separat mitgeteilt)
username	(Ihr Benutzername für die Testumgebung)	(Ihr Benutzername)
password	(Ihr Passwort für die Testumgebung)	(Ihr Passwort)
scope	openid	openid
grant_type	password	password

Sie erhalten in der Antwort einen Access-Token, welcher ein JSON Web Token ist (JWT; <https://jwt.io/>). Dieser Access-Token ist nur für eine bestimmte Zeit gültig. Mithilfe eines JWT-Parsers, welcher häufig in Software-Bibliotheken zu finden ist, kann aus dem Token der Zeitpunkt, bis wann dieser gültig ist, herausgelesen werden. Wenn der Token nicht mehr gültig ist, muss mit derselben API ein neuer Token angefordert werden.

**Der Access-Token muss in jedem Request an DESH im Header im Feld Authorization mitgegeben werden.**

Erreichbar ist die DESH-Anwendung unter der URL:

<https://desh.bdr.de>

Die Testumgebung ist unter der folgenden URL erreichbar:

<https://demo.desh.bdr.de>.

## 2.2 Schritte zur Bereitstellung

### 2.2.1 Schritt 1: Tan Generator

Die erste Schnittstelle wird genutzt, um am Tan Generator TANs anzufordern. Bei der TAN handelt es sich um eine UUID V4. Diese TANs werden über einen Service der Bundesdruckerei generiert. Diese TANs werden zurückgesandt. Die übertragenen Daten werden nicht gespeichert. Daher kann die TAN auch eigenständig durch das Empfängersystem generiert werden.

Anschließend setzt das Empfängersystem die `demis_transaction_id` zusammen.

Format: IMS-12345-CVDP-XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX

IMS: festes Prefix

12345: 5-stelliger Identifier des sequenzierenden Labors (Untersuchungslabors), analog zum bereits existierenden DEMIS-System (DEMIS-10001 bis derzeit DEMIS-10563). Die Liste wird von der DEMIS-Geschäftsstelle gepflegt.

CVDP: 4-stelliges DEMIS-Kürzel, welches dem Meldetatbestand direkt zugeordnet wird.

XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX: automatisch vom System generierte TAN.

### 2.2.2 Schritt 2: Übermittlung der Daten

Die zweite Schnittstelle wird genutzt, um die Meta- und Sequenzdaten an die Bundesdruckerei zu übertragen. Dort werden diese Informationen gespeichert und für das RKI zur Verfügung gestellt.

## 3 Spezifikation

Die Spezifikation nach OpenAPI 3.0.3 wird separat über die mitgelieferte Datei „receiver.yaml“ definiert. In dieser können Sie alle möglichen Request nachvollziehen und gegen die Testumgebung ausprobieren, wenn Sie einen entsprechenden Editor (z. B. <https://editor.swagger.io/>) nutzen und das TLS-Zertifikat installiert ist.

Folgend werden die verschiedenen Abschnitte der Spezifikation näher erläutert.

### 3.1 Authentifizierung

In diesem Abschnitt ist die Request-Spezifikation hinterlegt, um einen Access-Token anzufordern (siehe dazu Kapitel 2.1).

### 3.2 Lookups

Unter „Lookups“ werden Listen verstanden, die genutzt werden, um bestimmte Werte zu validieren. Es gibt einen Request, um alle hinterlegten Lookups abzufragen sowie die Requests zum Abfragen der Listen/Werte.

Aktuell gibt es folgende Listen:

1. **reporting\_lab**  
Liste mit allen validen Labornummern von Laboren, die sequenzieren
2. **sample\_type**  
Liste mit allen validen Stellen zur Probeentnahme
3. **sending\_lab**  
Liste mit allen validen Labornummern von Laboren, die Proben entnehmen

**4. seq\_type**

Liste mit allen validen Sequenzierungsverfahren

**5. seq\_reason**

Liste mit allen validen Sequenzierungsgründen

### 3.3 Data

Siehe ergänzend zu der folgenden Beschreibung auch Kapitel Kapitel 2.2.2.

In diesem Abschnitt ist der Request zu finden, um die genaue Spezifikation der zu übermittelnden Daten nachzuvollziehen. Die Antwort beinhaltet die identifizierenden Datenfelder und die medizinischen Datenfelder sowie die Beschreibung der Validierungen für die Datenfelder.

In diesem Abschnitt ist der Request zu finden, um die pseudonymisierten Daten zu verschicken.

Neben dem Senden einzelner Datensätze ist es ebenfalls möglich mehrere Datensätze mittels eines Requests zu schicken. Dabei muss in dem Request-Body ein Array von Datensätzen vorhanden sein. Jedem Datensatz kann eine ID gegeben werden (Feld: **\_id**). Diese ID wird mit einer entsprechenden Fehlermeldung zurückgegeben, falls der Datensatz mit dieser ID einen Fehler hat. Die ID hat keinen festen Typ.

Die Calls und deren Aufbau sind der Datei „receiver.yaml“ zu entnehmen.

