

Dokumentation GeDaTrans-API v1.0.0

Ansprechpartner dena

- Henri Lukas
- E-Mail: lukas@dena.de
- Telefon: 030-726165762

Zweckbeschreibung

Der Zweck des API (Application Programming Interface: Schnittstelle) liegt in der automatischen Übertragung von Daten

aus proprietärer Software in den GeDaTrans der Deutschen Energie-Agentur (dena).

Aktuell stehen folgende Funktionen bereit:

1. Praxisnachweis

Standard-Fehlercodes

- **0** - Anfrage erfolgreich, alles in Ordnung
- **10** - falscher Request-Typ
- **20** - fehlende Daten
- **21** - Daten sind nicht valide
- **99** - sonstiger Fehler

Zuordnung API-Fehlercode (links) zu HTTP-Status-Codes (rechts)

- **0** ---- **200 OK**
- **10** ---- **405 Method not allowed**
- **20** ---- **400 Bad Request**
- **21** ---- **400 Bad Request**
- **99** ---- **500 Internal Server Error**

Standard-Dispatcher-IDs

- der Dispatcher ist der Einstiegspunkt in das API für die Weiterverarbeitung von Daten
- ein direkter Zugriff auf GeDaTrans-Module (z. B. Praxisnachweis-Login) ist nicht zulässig
- es ist ausschließlich der hier genannte Link <https://www.gedatrans.de/dispatcher/> für die Weiterverarbeitung der Daten zu nutzen
- Hinweise zur Parameterisierung erfolgen in den einzelnen Schnittstellenbeschreibungen
- folgende Services sind implementiert:
 - **16** ---- Praxisnachweis (<https://www.gedatrans.de/dispatcher/16/%TOKEN%>)

Praxisnachweis

Kurzbeschreibung

Bereitstellung von Daten für das GeDaTrans-Modul *Praxisnachweis*.

Für den Transfer ist ein Login im GeDaTrans notwendig.

Die übermittelten Daten müssen im Format gd2 vorliegen und werden automatisch gegen ein xsd-Schema validiert.

Anfrage: *Import initialisieren*

URL

- <https://www.gedatrans.de/api/transfer/request-import>

Typ

- **POST**

Aufbau der Anfrage

Der Anfrage-Body besteht aus zwei POST-Variablen:

- **gd2** : Die Daten im gd2-Format (komplette XML-Struktur) als String
- **meta** : Ein JSON-String mit zusätzlichen Informationen zur Anfrage
 - im meta-Feld sollte mindestens die Software-Version mitgeteilt werden

- für die Implementierung ist das Objekt `software` mit den beiden Schlüsseln `name` und `version` zu nutzen
- siehe folgendes Beispiel

Beispiel-Anfrage-Body

```
gd2: <?xml version="1.0"?><gedatrans version="1.0.2"><praxisnachweis><leistungsnachweis_typ> ...  
meta: {"software":{"name":"Test-Software","version":"1.15.12"}}
```

Beispiel-Anfrage-Body (PHP)

```
$_POST = [  
    'gd2' => '<?xml version="1.0"?><gedatrans version="1.0.0"><praxisnachweis><leistungsnachweis_typ> ...'  
    'meta' => '{"software":{"name":"Test-Software","version":"1.15.12"}'  
]
```

Beispiel-Header der Anfrage

```
Accept: */*  
Accept-Encoding: gzip, deflate  
Accept-Language: de-DE, de; q=0.8, en-US; q=0.6, en; q=0.4  
Cache-Control: no-cache  
Connection: keep-alive  
Content-Length: 6630  
Content-Type: application/x-www-form-urlencoded  
Cookie: PHPSESSID=amikiuk5l1trh1ft1c93780if72  
Host: example.gedatrans.de  
Origin: http://example.gedatrans.de  
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0 ...
```

Rückgabe

Die Rückgabe ist in drei Objekte geteilt:

`error` gibt Auskunft über den Bearbeitungsstand der gesendeten Daten und besteht aus einem Fehler-Code und einer Nachricht

data enthält die zurückgelieferten Daten (kann auch leer sein, bspw. im Fehlerfall)

meta liefert Informationen über den anfragenden Client, etc.

- Erfolg

```
{
  "error": {
    "code": 0,
    "message": "Data accepted - token generated"
  },
  "data": {
    "token": "3Murnbqmyo0KEjIbEjKA02hhIWIEWS-U",
    "url": "http://frontend.gedatrans.kv85/dispatcher/16/3Murnbqmyo0KEjIbEjKA02hhIWIEWS-U"
  },
  "meta": {
    "request_ip": "192.168.5.111",
    "request_time": 1443528643
  }
}
```

- falscher Request-Type

```
{
  "error": {
    "code": 10,
    "message": "This request requires method POST only"
  },
  "data": {},
  "meta": {
    "request_ip": "192.168.5.111",
    "request_time": 1443517975
  }
}
```

- fehlende Daten

```
{
  "error": {
    "code": 20,
    "message": "No data present"
  },
  "data": {},
  "meta": {
    "request_ip": "192.168.5.111",
    "request_time": 1443518062
  }
}
```

- gd2 nicht valide

```
{
  "error": {
    "code": 21,
    "message": "The gd2 is not valid. Please check the xsd schema file"
  },
  "data": {},
  "meta": {
    "request_ip": "192.168.5.111",
    "request_time": 1443518171
  }
}
```

- sonstiger Fehler

```
{
  "error": {
    "code": 99,
    "message": "Critical Error. Please contact administrator"
  },
  "data": {},
  "meta": {
    "request_ip": "192.168.5.111",

```

```
    "request_time": 1443518247
  }
}
```

Programm-Ablauf

- Software übermittelt die Daten in gd2 an das API
- API prüft auf formale Korrektheit der Anfrage
- API validiert die gd2-Daten
- im Erfolgsfall liefert API einen Token

Import der Daten in GeDaTrans

- Software öffnet Browser mit folgendem Link: <https://www.gedatrans.de/dispatcher/16/%TOKEN%>
 - **%TOKEN%** wird durch das eigentliche Token ersetzt
 - Beispiel: https://www.gedatrans.de/dispatcher/16/6yuA3HMzR_c9IjdPjk6-mkIrmjFKRnUh
- dieser Link wird zusätzlich noch in der Rückgabe mitgeliefert
- bei Gültigkeit des Tokens und erfolgreichem Login werden die Daten importiert

Spezifizierung Token

- Zeichenkette
- 32 Zeichen
- Zeichenvorrat: A-Za-z0-9_-
- Gültigkeit: 15 Minuten
- Beispiel: [xPq3mWlLDMoZZNWztzFU9beyWuCv93h9](#)

Die Rückgabe für alle Anfragen unabhängig von Erfolg oder Misserfolg ist ausschließlich JSON.

Changelog

v1.0.0 - 2015-09-30

- initiale Veröffentlichung
- Funktionen:
 - Import Praxisnachweis