



Der Token-Endpoint antwortet mit einem JWT-Zugriffstoken:

```
1 {"token": "YOUR_ACCESS_TOKEN"}
```

Ihre Anwendung muss dann den Token extrahieren und diesen als Bearer-Token an die Endpunkte senden, die sie aufrufen möchte.

Beispiel Curl-Anfrage an einen Endpunkt:

```
1 curl --location 'https://qa-api.obelis-deutschlandnetz.de/ocpi/bob/2.2/debob-reports/{endpoint}' \
2 --header 'Content-Type: application/json' \
3 --header 'Authorization: Bearer YOUR_ACCESS_TOKEN'
```

Alle Zugriffstoken sind eine Stunde lang gültig. Der Token selbst enthält die Ablaufzeit, die manuell über  [JWT.IO](https://jwt.io) überprüft werden kann oder automatisch mithilfe eines JWT-Pakets in Ihrer bevorzugten Programmiersprache überprüft werden kann.

Nachdem der Token abgelaufen ist, muss die Anwendung einen neuen Token anfordern.

PAYLOAD: DATA

```
{
  "sub": "33nrpagk6sfj7ngt8tojf3gv9",
  "token_use": "access",
  "scope": "transactions/m2m",
  "auth_time": 1695114470,
  "iss": "https://cognito-idp.eu-central-1.amazonaws.com/eu-central-1_KNrpR8om1",
  "exp": 1695118070, Tue Sep 19 2023 12:07:50 GMT+0200 (Mitteleuropäische Sommerzeit)
  "iat": 1695114470,
  "version": 2,
  "jti": "6c6bb550-a532-4f15-be9c-4e7e0d84b21f",
  "client_id": "33nrpagk6sfj7ngt8tojf3gv9"
}
```

Beispiel für die Daten eines Tokens, die das Ablaufdatum zeigen

Alle API-Aufrufe müssen über [HTTPS](https://) erfolgen. Aufrufe über einfaches HTTP werden fehlschlagen.

API-Anfragen ohne Authentifizierung werden ebenfalls fehlschlagen.

## Fehlermeldungen

OBELISdeutschlandnetz verwendet herkömmliche HTTP-Antwortcodes, um den Erfolg oder das Scheitern einer API-Anfrage anzuzeigen. Im Allgemeinen gelten folgende Regeln: Codes im Bereich `2xx` zeigen Erfolg an. Codes im Bereich `4xx` zeigen einen Fehler an, der aufgrund der bereitgestellten Informationen fehlgeschlagen ist (z. B. ein erforderlicher Parameter wurde ausgelassen, eine Gebühr konnte nicht durchgeführt werden, usw.). Codes im Bereich `5xx` zeigen einen Fehler mit den Servern von OBELISdeutschlandnetz an (diese sind selten).

Einige 4xx-Fehler, die programmgesteuert behandelt werden können (z. B. ein ungültiges Datumsfeld), enthalten einen Fehlercode, der den gemeldeten Fehler kurz erklärt.

## HTTP Status Code Übersicht

Statuscode	Beschreibung
200 - OK	Alles hat wie erwartet funktioniert.
400 - Bad Request	Die Anfrage war nicht akzeptabel, oft aufgrund eines fehlenden erforderlichen Parameters.
401 - Unauthorized	Es wurde kein gültiges OAuth 2.0-Token bereitgestellt.
403 - Forbidden	Die Aktion ist nicht erlaubt.
404 - Not Found	Die angeforderte Ressource existiert nicht.
429 - Too many Requests	Zu viele Anfragen haben die API zu schnell erreicht. Wir empfehlen, Ihre Anfragen exponentiell zu verzögern.
500, 502, 503, 504 - Server Errors	Etwas ist auf OBELISdeutschlandnetz's Seite schiefgelaufen. (Diese sind selten.)

Beispiel eines Fehlers (der Statuscode ist optional):

```
1 {"statusCode": 400, "message": "Charging Hub already exists for this coordinates"}
```

## Rate Limiting

Die OBELISdeutschlandnetz API verwendet verschiedene Schutzmaßnahmen gegen plötzliche Anstiege des eingehenden Datenverkehrs, um die Stabilität zu maximieren. Anwendungen, die zu viele Anfragen in schneller Abfolge senden, können Fehlerantworten erhalten, die als Statuscode 429 angezeigt werden. In der API gibt es verschiedene Begrenzungen, einschließlich eines **Rate Limiting**, die die Anzahl der innerhalb einer Sekunde empfangenen Anfragen begrenzt:

**OBELISdeutschlandnetz erlaubt bis zu 2.000 Operationen pro Minute.**

Behandeln Sie diese Begrenzung als Maximum und generieren Sie keine unnötige Last. Weitere Informationen zur Behandlung von 429er-Fehlern finden Sie im Abschnitt "Elegantes Behandeln von Begrenzungen". Wenn Sie plötzlich eine steigende Anzahl von Anfragen mit Ratenbegrenzung feststellen, [kontaktieren](#) Sie bitte den Support.

Wir können die Begrenzungen reduzieren, um Missbrauch zu verhindern, oder erhöhen, um Anwendungen mit hohem Datenverkehr zu unterstützen.

## Häufige Ursachen und Maßnahmen

Ratenbegrenzungen können unter verschiedenen Bedingungen auftreten, sind jedoch am häufigsten in diesen Szenarien:

- Das Ausführen einer großen Anzahl von eng aufeinander folgenden Anfragen kann zu Ratenbegrenzungen führen. Dies ist oft Teil einer analytischen oder Migrationsoperation. Wenn Sie an solchen Aktivitäten teilnehmen, sollten Sie versuchen, die Anfragegeschwindigkeit auf der Anwendungsseite zu steuern (siehe "Elegantes Behandeln von Begrenzungen").
- Ein plötzlicher Anstieg des Datenvolumens kann zu Ratenbegrenzungen führen. Wir versuchen, unsere Raten hoch genug festzulegen, damit die überwiegende Mehrheit der Benutzer niemals aufgrund legitimen Datenverkehrs begrenzt wird. Wenn Sie jedoch vermuten, dass ein bevorstehendes Ereignis Sie über die oben aufgeführten Grenzen bringen könnte, [kontaktieren](#) Sie bitte den Support.

## Elegantes Behandeln von Begrenzungen

Eine grundlegende Technik für Integrationen, um Begrenzungen elegant zu behandeln, besteht darin, nach Statuscodes 429 zu suchen und einen Wiederholungsmechanismus zu implementieren. Der Wiederholungsmechanismus sollte einen exponentiellen Backoff-Zeitplan befolgen, um die Anzahl der Anfragen zu reduzieren, wenn dies erforderlich ist. Es wird auch empfohlen, in den Backoff-Zeitplan etwas Zufälligkeit einzubauen, um den Effekt einer "thundering herd" zu vermeiden.

Sie können einzelne Anfragen nur in begrenztem Maße optimieren. Eine noch ausgefeiltere Methode wäre daher, den Datenverkehr zu OBELISdeutschlandnetz auf globaler Ebene zu steuern und ihn bei erheblicher Ratenbegrenzung zu reduzieren. Eine häufige Technik zur Steuerung der Raten ist die Implementierung eines [Token-Bucket-Ratenbegrenzungsalgorithmus](#) auf der Anwendungsseite. Fertige und ausgereifte Implementierungen für Token Bucket sind in fast jeder Programmiersprache verfügbar.

## Lasttest

Es ist üblich, dass Benutzer:innen sich auf ein großes Ereignis vorbereiten, indem sie ihre Systeme mit der OBELISdeutschlandnetz-API als Teil davon einer Belastungsprüfung unterziehen. Wir raten im Allgemeinen von dieser Praxis ab, da die API-Limits im Testmodus niedriger sind, sodass die Belastungsprüfung auf Limits stoßen kann, die im Produktionsmodus nicht erreicht werden. Der Testmodus ist auch kein perfekter Ersatz für Live-API-Aufrufe und kann daher irreführend sein. Beispielsweise können Anfragen an unsere Testumgebung zu deutlich unterschiedlichen Latenzprofilen führen als Anfragen an unsere Live-Umgebung.

Als Alternative empfehlen wir, Integrationen so zu erstellen, dass sie über ein konfigurierbares System zum Simulieren von Anfragen an die OBELISdeutschlandnetz-API verfügen, das für Belastungsprüfungen aktiviert werden kann. Um realistische Ergebnisse zu erzielen, sollten Sie die Latenz simulieren, indem Sie für eine Zeit in den Schlafmodus gehen, die Sie durch das Abtasten der Dauer realer API-Aufrufe festlegen, wie sie aus der Sicht der Integration gesehen werden.

## API Dokumentation

Wir bitten Sie, vor einer persönlichen Anfrage zunächst immer die [Swagger](#)- bzw. [OpenAPI](#) YAML-Beschreibungen der OBELISdeutschlandnetz-API eingehend zu sichten.

Diese beiden technischen Dokumentationen werden fortlaufend aktualisiert und sind als abschließend zu betrachten, d.h. dort nicht aufgeführte Funktionen und Abfragemöglichkeiten sind zum gegenwärtigen Zeitpunkt auch (noch) nicht verfügbar.

Bitte haben Sie zudem Verständnis, dass Informationen zu Aspekten mit Einfluss auf die Gesamtstabilität und Verfügbarkeit der Anwendung sowie der API (bspw. Konfiguration der Firewall), welche über die in diesem Dokument verfügbaren Angaben hinausgehen, auch auf Nachfrage nicht veröffentlicht bzw. kommuniziert werden können.

## Verwendung der OBELISdeutschlandnetz API

Die OBELISdeutschlandnetz API ermöglicht es Ihnen, auf verschiedene Ressourcen zuzugreifen und damit zu interagieren. In dieser Sektion erfahren Sie, wie Sie generierte Ressourcen identifizieren, speichern und aktualisieren können.

### Erstellung von Ressourcen

Um eine neue Ressource zu erstellen, verwenden Sie den HTTP-Verb `POST`. Stellen Sie sicher, dass Sie die erforderlichen Daten und Felder in Ihrem Anfragekörper bereitstellen. Die API wird Ihnen dann eine eindeutige Identifikation (ID) für die erstellte Ressource zurückgeben.

Hier ist ein Beispiel für die Erstellung einer Ladestation:

```
1 POST https://qa-api.obelis-deutschlandnetz.de/ocpi/bob/2.2/debob-reports/charginghubs
2 Content-Type: application/json
3 Authorization: Bearer YOUR_ACCESS_TOKEN
4
5 {
6   "name": "Charging Hub",
7   "address": "Berliner street",
8   "postalCode": "13465",
9   "city": "Berlin",
10  "state": "BE",
11  "coordinates": {
12    "latitude": 52.643,
13    "longitude": 13.287
14  },
15  "operator": "1230P",
16  "owner": "Allego GmbH",
17  "negligibleDamages": true,
18  "lastUpdated": "2023-07-24T13:00:58.583Z"
19 }
```

In dieser Anfrage erstellen Sie eine neue Ladestation mit den angegebenen Daten in den Feldern.

Die API wird Ihnen mit einer ID für diese neue Ressource antworten, die Sie für zukünftige Interaktionen verwenden können.

```
1 {
2   "chargingHubId": "36e69843-55f1-4513-8e2d-dba209ccc03c"
3 }
```

In der OBELISdeutschlandnetz API sind Ressourcen ein zentraler Bestandteil. Jede Ressource wird durch einen eindeutigen Identifikator (ID) repräsentiert, der Ihnen bei der Erstellung der Ressource zugeordnet wird. Dieser Identifikator ist entscheidend, um auf die Ressource zuzugreifen, sie zu aktualisieren oder zu löschen. Stellen Sie sicher, dass Sie den generierten Identifikator sorgfältig aufbewahren, da er zur eindeutigen Identifizierung der Ressource benötigt wird.

### Aktualisierung von Ressourcen

Um eine vorhandene Ressource zu aktualisieren, verwenden Sie die folgenden HTTP-Verben, abhängig von den Anforderungen Ihrer Aktualisierung:

- **PUT (Vollständige Aktualisierung):**

Bei Verwendung von `PUT` ersetzen Sie die gesamte Ressource durch die neuen Daten, die in der Anfrage bereitgestellt werden. Sie müssen die vollständige Repräsentation der Ressource, einschließlich aller Felder, in Ihrer Aktualisierungsanfrage senden. Dadurch wird die vorhandene Ressource durch die neuen Daten vollständig ersetzt.

```
1 PUT https://qa-api.obelis-deutschlandnetz.de/resource/{RESOURCE_ID}
2 Content-Type: application/json
3 Authorization: Bearer YOUR_ACCESS_TOKEN
4
5 {
6   "field1": "new_value1",
7   "field2": "new_value2"
8 }
```

In dieser Pseudoanfrage wird die Ressource mit der ID `{RESOURCE_ID}` aktualisiert und vollständig ersetzt.

Bitte beachten Sie, dass Sie den gesamten Ressourceninhalt in Ihrer Aktualisierungsanfrage senden müssen. Das bedeutet, dass Sie alle erforderlichen Daten zusammen mit den zu aktualisierenden Informationen senden müssen. Wenn Sie beispielsweise nur bestimmte Teile der Ressource aktualisieren möchten, müssen Sie den `PATCH`-Verb verwenden und nur die relevanten Felder senden.

- **PATCH (Teilweise Aktualisierung):**

Bei Verwendung von `PATCH` führen Sie partielle Aktualisierungen an der Ressource durch. Sie müssen nur die Felder senden, die Sie aktualisieren möchten, zusammen mit ihren neuen Werten. Dies ermöglicht genauere und teilweise Aktualisierungen der Ressource.

```
1 PATCH https://qa-api.obelis-deutschlandnetz.de/resource/{RESOURCE_ID}
2 Content-Type: application/json
3 Authorization: Bearer YOUR_ACCESS_TOKEN
4
5 {
6   "field1": "new_value1"
7 }
```

In dieser Anfrage wird nur das Feld `field1` der Ressource mit der ID `{RESOURCE_ID}` auf den neuen Wert `new_value1` aktualisiert, während alle anderen Felder unverändert bleiben.

Die [Swagger](#)- bzw. [OpenAPI](#)-Dokumentation gibt Auskunft darüber, welche Verben für einen bestimmten Endpunkt unterstützt werden. Dies ist wichtig, da nicht alle Endpunkte unbedingt beide Verben unterstützen.

## Aktualisierung von Ressourcen mit Unterentitäten

### **Achtung**

Bei der Aktualisierung von Ressourcen, die Arrays von Unterentitäten enthalten, sollten Sie vorsichtig sein. Wenn diese Unterentitäten keine eindeutigen Identifikatoren haben und Sie eine Aktualisierung an dieses Feld mit nur einer Entität senden, werden alle anderen Elemente im Array gelöscht.

Zur Veranschaulichung:

Wenn Sie die Konnektoren eines Ladepunkts aktualisieren möchten und Sie senden nur einen neuen Konnektor an den Endpunkt, ohne die vorhandenen Konnektoren zu erwähnen, werden alle bestehenden Konnektoren durch den Konnektor ersetzt. Um einzelne Konnektoren zu aktualisieren, müssen Sie alle aktuellen Konnektoren zusammen mit dem aktualisierten Konnektor senden, um sicherzustellen, dass keine Daten verloren gehen.

Stellen Sie sicher, dass Sie die API-Dokumentation und die Struktur der Ressource sorgfältig überprüfen, um sicherzustellen, dass Sie die erforderlichen Daten für die Aktualisierung von Ressourcen mit komplexen Strukturen korrekt senden.

Wenn Sie weitere Fragen zur Aktualisierung von Ressourcen mit komplexen Strukturen haben, zögern Sie nicht, sich an unser Support-Team zu wenden. Wir stehen Ihnen gerne zur Verfügung, um Ihre Fragen zu klären und Unterstützung zu bieten.

## Löschen von Ressourcen

Um eine Ressource zu löschen, verwenden Sie den HTTP-Verb `DELETE` und geben Sie den Identifikator der zu löschenden Ressource im Endpunkt an. Hier ist ein Beispiel für eine Löschungsanfrage:

```
1 DELETE https://qa-api.obelis-deutschlandnetz.de/resource/{RESOURCE_ID}
2 Authorization: Bearer YOUR_ACCESS_TOKEN
```

Diese Anfrage entfernt die Ressource mit der ID `{RESOURCE_ID}` aus dem System.

Seien Sie vorsichtig bei Löschvorgängen, da gelöschte Ressourcen nicht wiederhergestellt werden können.

## Abfrage kürzlich geänderter Ressourcen

Eine Entität gilt als geändert, wenn entweder diese selbst oder eine direkt verknüpfte Entität geändert wurden. Eine Aktualisierung bedeutet jedoch nicht unbedingt, dass sich die Felder einer Entität tatsächlich bereits geändert haben. Sollen die Werte der Datenbank regelmäßig oder in sehr kurzen Abständen aktualisiert werden, ist zu beachten, dass es einen Zeitversatz von bis zu ca. 15 Minuten geben kann, bis eine Änderung in der API sichtbar ist. In der Zwischenzeit kann es vorkommen, dass die API z.B. mit einem 404-Fehler antwortet.

## Monatliche Reports

Bitte beachten Sie, dass das Erstellen von monatlichen Berichten (Ratings) nur für den vorherigen Monat möglich ist. Dies bedeutet, dass Sie Berichte nur für Daten im vorherigen Monat erstellen können. Stellen Sie sicher, dass Ihre Anfragen diese zeitliche Begrenzung berücksichtigen.

Diese Validierung betrifft folgende Endpunkte:

- /ocpi/bob/2.2/debob-reports/hotline-ratings
- /ocpi/bob/2.2/debob-reports/nondiscriminatory-access
- /ocpi/bob/2.2/debob-reports/charginghub/{charginghubId}/ratings
- /ocpi/bob/2.2/debob-reports/charginghub/{charginghubId}/ad-hoc-tariff
- /ocpi/bob/2.2/debob-reports/charginghub/{charginghubId}/chargingpoint/{chargingpointId}/ratings