

**Dokumentation
ISPSuite DNS-API**

ab API-Version 20.9.1

**Letzte Aktualisierung des Dokuments:
Dienstag, 21. Februar 2023**

1.1.	API-Schlüssel	3
1.2.	Schlüsselberechtigungen	3
2.	Nutzung der API	4
2.1.	JSON-Basisobjekt.....	5
2.2.	rrset-Objekt.....	5
2.3.	JSON-Beispielobjekte.....	6
2.3.1.	Hinzufügen eines A-Records.....	6
2.3.2.	Ersetzen eines A-Records.....	6
2.3.3.	Löschen eines A-Records	7
2.4.	Beispielaufruf (curl).....	8
2.5.	Code-Beispiel (PHP)	9
2.6.	Rückgabewerte der API.....	10
2.6.1.	Mögliche Fehlermeldungen in „API_Comment“	10
2.6.1.1.	You have no access to this name.....	10
2.6.1.2.	Missing Parameters.....	10

Die ISPSuite bietet Ihnen die Möglichkeit, DNS-Einträge per API abzurufen und zu verwalten.

1.1. API-Schlüssel

Hinweis: Um die API nutzen zu können, müssen API-Zugangsschlüssel für die jeweilige DNS-Zone angelegt werden. Dies ist aktuell nicht in jedem Tarif möglich. Bitte erkundigen Sie sich ggf., ob die Nutzung der API in dem von Ihnen gebuchten Tarif möglich ist.

1.2. Schlüsselberechtigungen

Über Zugriffsregeln kann jedem API-Schlüssel eine individuelle Berechtigung für den Zugriff auf Zweige der DNS-Zone gewährt werden, um z.B. nur den Zugriff auf bestimmte Subdomain-Namen zuzulassen. Das Sternchen (*) steht als Wildcard-Operator für beliebig viele Elemente, jedoch mindestens eines, im Zweig und kann als Präfix, Infix und Suffix verwendet werden.

Zugriffs-Regel	
*	Erlaubt den Zugriff auf die gesamte DNS-Zone
*.domain.tld	Erlaubt den Zugriff auf beliebige Subdomains von domain.tld, nicht jedoch domain.tld selbst.
_*acme-challenge.*	Erlaubt an beliebiger Stelle im DNS-Zweig den Zugriff auf den Namen „_acme-challenge“ (z.B. zur Verwendung mit Let's Encrypt)
*.subdomain.domain.tld	Erlaubt den Zugriff auf Subdomains unterhalb von „subdomain.domain.tld“, sowie den Namen selbst.
www.*.subdomain.domain.tld	Erlaubt Zugriff auf den Namen „www“ in beliebiger Tiefe unterhalb der Subdomain „subdomain.domain.tld“
subdomain.domain.tld	Erlaubt ausschließlich den Zugriff auf den Namen „subdomain.domain.tld“, nicht aber auf andere darunter oder darüber liegende Namen im Zweig.

2. Nutzung der API

Die API ist unter folgender URL erreichbar:

<https://wr.ispsuite.portunity.de/api/>

Für alle Anfragen müssen (wenn nicht anders angegeben) immer mindestens folgende Parameter als JSON-Objekt im via POST-Request im Parameter „data=“ übergeben werden.

Die Authentifizierung erfolgt per Basic-Auth mit dem Tupel aus Produkt-Benutzername und API-Schlüssel.

Parameter	Beschreibung	Beispiel
api	Der Name der zu verwendenden API. Für die DNS-API immer „product-dns“	product-dns
action	Die auszuführende Aktion: <ul style="list-style-type: none">• get Abrufen der DNS-Zone• add Hinzufügen von DNS-Records• replace Ersetzen vorhandener DNS-Records• delete Löschen vorhandener DNS-Records	Siehe ausführliche Beispielliste in Abschnitt 2.3
rrset	Nur bei add/replace/delete: Das rrset-Objekt, welches die betroffenen DNS-Records darstellt.	Siehe Ausführliche Beschreibung in 2.2

Tabelle 1

2.1. JSON-Basisobjekt

Alle Anfragen müssen per POST gestellt werden und müssen als Parameter „data“ mindestens folgendes JSON-Objekt enthalten:

```
{
  "api": "product-dns",
  "action": "get/add/replace/delete"
}
```

Code-Snippet 1

Der Wert von „action“ muss hierbei natürlich durch die auszuführende Aktion ersetzt werden.

2.2. rrset-Objekt

Für die Aktionen add, replace, delete muss zudem ein rrset-Objekt ins Basisobjekt eingefügt werden:

```
"rrset": {
  "type": "AAAA",
  "name": "beispiel-subdomain.domain.tld",
  "ttl": 14400,
  "www": false,
  "records": [
    {
      "content": "2001:db8::1"
    },
    {
      "content": "2001:db8::2"
    }
  ]
}
```

Code-Snippet 2

Feldname	Beschreibung	Beispiel / Gültige Werte
type	Der DNS RR-Typ	A / AAAA / MX / SRV / TXT / CNAME / NS
name	Der vollständige FQDN, welcher bearbeitet werden soll	domain.tld subdomain.domain.tld
ttl	TTL in Sekunden	30 3600 (bis zu 2 ³² -1)
www	Optional, Boolean: Soll eine Kopie des Eintrags zusätzlich mit „www.“-Präfix angelegt werden?	true <u>false</u> (Standard)
records	Ein Array, welches die zu bearbeitenden Record-Inhalte enthält, jeweils aufgebaut als „content => value“-Paare	content => „Beispieltext“ content => „1.2.3.4“

Tabelle 2

Alle hier gelisteten Parameter außer „www“ müssen immer im rrset-Objekt vorhanden sein!

2.3. JSON-Beispielobjekte

2.3.1. Hinzufügen eines A-Records

Mit diesem Objekt wird ein A-Record mit der IP-Adresse 203.0.113.1 hinzugefügt. Eventuell bestehende Records mit gleichem Namen und Typ werden dabei nicht ersetzt, der neue Record wird in dem Fall als zusätzlicher Record angefügt.

```
{
  "api": "product-dns",
  "action": "add",
  "rrset": {
    "type": "A",
    "name": "beispiel-subdomain.domain.tld",
    "ttl": 180,
    "www": false,
    "records": [
      {
        "content": "203.0.113.1"
      }
    ]
  }
}
```

Code-Snippet 3

2.3.2. Ersetzen eines A-Records

Mit diesem Objekt werden alle bestehenden A-Records mit dem Namen „beispiel-subdomain.domain.tld“ durch diesen neuen Record ersetzt – unabhängig von der Anzahl der Records in diesem Objekt und der Anzahl bereits vorhandener Records. Ist kein A-Record vorhanden, wird er neu angelegt.

```
{
  "api": "product-dns",
  "action": "replace",
  "rrset": {
    "type": "A",
    "name": "beispiel-subdomain.domain.tld",
    "ttl": 180,
    "www": false,
    "records": [
      {
        "content": "203.0.113.2"
      }
    ]
  }
}
```

Code-Snippet 4

2.3.3. Löschen eines A-Records

Mit diesem Objekt wird der A-Record mit dem Wert „203.0.113.2“ gelöscht.
Andere A-Records mit gleichem Namen, aber anderem Inhalt bleiben davon unberührt.
Ist kein Record mit dem Tupel aus angegebenem Namen, Typ und Wert vorhanden,
werden keine Einträge gelöscht.

Tipp 1: Falls Sie den Record löschen möchten, um einen neuen anzulegen, ist die „replace“-Action (Beschrieben in 2.3.2) für Ihren Anwendungsfall eventuell besser geeignet.

Tipp 2: Rufen Sie ggf. vorher mit „get“ den Inhalt der Zone ab um zu ermitteln, welche Einträge vorhanden sind.

```
{
  "api": "product-dns",
  "action": "delete",
  "rrset": {
    "type": "A",
    "name": "beispiel-subdomain.domain.tld",
    "ttl": 180,
    "www": false,
    "records": [
      {
        "content": "203.0.113.2"
      }
    ]
  }
}
```

Code-Snippet 5

2.4. Beispielaufruf (curl)

Mit folgendem curl-Aufruf wird ein A-Record für „beispiel-subdomain.domain.tld“ mit dem Wert „203.0.113.1“ angelegt:

```
$ curl -v 'https://wr.ispsuite.portunity.de/api/' \  
  --basic \  
  --user 'dns12345:0821d880aabaf5d69f30c5e2baa915b8' \  
  --data 'data=  
{  
  "api": "product-dns",  
  "action": "replace",  
  "rrset": {  
    "type": "A",  
    "name": "beispiel-subdomain.domain.tld",  
    "ttl": 180,  
    "www": false,  
    "records": [  
      {  
        "content": "203.0.113.1"  
      }  
    ]  
  }  
}'
```

Code-Snippet 6

2.5. Code-Beispiel (PHP)

```
$apiUrl = 'https://wr.ispsuite.portunity.de/api/';

$dnsData = [
    'api' => 'product-dns',
    'action' => 'add',
    'rrset' => [
        'type' => 'A',
        'name' => 'beispiel-subdomain.domain.tld',
        'ttl' => 180,
        'www' => false,
        'records' => [
            ['content' => '203.0.113.1'],
        ],
    ],
];

$ch = curl_init();

curl_setopt_array($ch,
    [
        CURLOPT_URL => $apiUrl,
        CURLOPT_USERPWD => 'dns12345:0821d880aabaj5d69f30c5e2baa915b8',
        CURLOPT_POST => true,
        CURLOPT_POSTFIELDS => 'data='.json_encode($dnsData),
        CURLOPT_RETURNTRANSFER => true,
    ]
);

echo( curl_exec($ch) );
```

Code-Snippet 7

2.6. Rückgabewerte der API

Sie erhalten von der API eine Rückgabe in Form eines JSON-Objekts:

```
{
  "API_RequestDateTime": "2020-10-28 20:45:50",
  "API_RequestIP": "172.17.0.101",
  "API_Status": "ok",
  "API_Version": "20.7.6",
  "API_Comment": "ggf. Fehlermeldungen"
}
```

„API_Status“ kann die Werte „ok“ oder „failed“ annehmen. Im Falle von „ok“ wurde die Bearbeitung der Anfrage erfolgreich abgeschlossen, im Falle von „failed“ sind Fehler aufgetreten, welche in „API_Comment“ näher spezifiziert sind.

Details zu möglichen Fehlermeldungen und deren Ursache finden Sie in Abschnitt 2.6.1.

2.6.1. Mögliche Fehlermeldungen in „API_Comment“

2.6.1.1. You have no access to this name

Dieser Fehler tritt auf, wenn Sie versuchen mit einem API-Schlüssel auf Elemente zuzugreifen, auf welche dieser Schlüssel keinen Zugriff hat.

Mögliche Ursachen sind:

- **Sie versuchen mit dem Schlüssel einer anderen DNS-Zone zuzugreifen.**
Für jede DNS-Zone wird ein eigener API-Schlüssel benötigt. Der API-Schlüssel für Domain A hat keinen Zugriff auf die Zone von Domain B, auch wenn sich beide im selben Kundenkonto befinden.
- **Sie versuchen auf einen Namen innerhalb des DNS-Zweigs zuzugreifen, für den der Schlüssel nicht berechtigt ist.**
Prüfen Sie, ob die Zugriffsrechte ausreichen, um den gewünschten Namen bearbeiten zu können. Details dazu finden Sie in Abschnitt 1.2.
- **Sie haben keinen FQDN als „name“ angegeben.**
Bitte stellen Sie sicher, dass Sie den vollständigen FQDN, welchen Sie bearbeiten möchten, auch im Request angegeben haben.
"name": "beispiel-subdomain" ist ungültig;
"name": "beispiel-subdomain.domain.tld" ist gültig.

2.6.1.2. Missing Parameters

Dieser Fehler tritt auf, wenn benötigte Parameter innerhalb des JSON-Anfrageobjekts fehlen oder fehlerhafte Werte haben. Informationen zu den mitzuliefernden Parametern finden Sie in Code-Snippet 1, Code-Snippet 2 sowie Tabelle 1 und Tabelle 2.