

Publisher Webservice

- [Verwaltung von Projekten über die V6 REST-API](#)
- [Publisher-Verwaltung über die V6 REST-API](#)

Verwaltung von Projekten über die V6 REST-API

Einleitung

Die V6 REST-API bietet Admins die Möglichkeit, Projekte effizient zu verwalten. Mit dem `/Projects`-Endpoint können Projekte erstellt, ausgelesen, bearbeitet und gelöscht werden. Zusätzlich können Projekte anhand verschiedener Parameter gefiltert werden.

Endpoint

Für Admins: `https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects`

Parameter & Filter

Erforderliche Parameter (bei POST und PUT)

Parameter	Typ	Beschreibung
<code>title</code>	String	Titel des Projekts (Pflichtfeld)
<code>status</code>	Integer	Status des Projekts (Pflichtfeld)
<code>url</code>	String	URL des Projekts (Pflichtfeld)

Filteroptionen (bei GET)

Es kann nach allen Spalten der Tabelle `publisher.projects` gefiltert werden, z. B.:

- `id`
- `status`
- `publisher_id`
- `projecttype`
- `title`
- `url`

- `channel_id`
- `hidden`

Beispiel für GET mit Filtern:

```
curl -X GET -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token: ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects?status=1&publisher_id=10"
```

CRUD-Operationen

GET: Abrufen von Projekten

Abrufen einer Liste von Projekten oder eines spezifischen Projekts anhand der `id`.

```
curl -X GET -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token: ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects?id=5"
```

POST: Erstellen eines neuen Projekts

```
curl -X POST -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token: ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" -d '{"title": "Neues Projekt", "status": 1, "url": "https://example.com"}' "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects"
```

PUT: Aktualisieren eines bestehenden Projekts

```
curl -X PUT -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token: ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" -d '{"id": 5, "title": "Geändertes Projekt", "status": 2, "url": "https://updated-example.com"}
```

```
}' "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects"
```

DELETE: Löschen eines Projekts

Löschen eines Projekts anhand der `id`.

```
curl -X DELETE -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token: ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects?id=5"
```

Antworten (Responses)

Erfolgreiche Antwort

```
{
  "id": 5,
  "status": 1,
  "publisher_id": 10,
  "projecttype": "page",
  "title": "Projektbeispiel",
  "description": "Beschreibung des Projekts",
  "url": "https://example.com",
  "channel_id": 2,
  "project_identifizier_string": "abc123",
  "reach": "1000",
  "statistic_type": "internal",
  "external_sources_entity_id": "-1",
  "trackingtemplate": "template",
  "kpi_whitelist": "kpi1,kpi2",
  "hidden": false,
  "login_type": "pub",
  "exclude_from_salary": 0,
  "default_project": false,
  "insert_timestamp": "2025-01-27T12:00:00"
}
```

Fehlerhafte Anfrage

```
{  
  "error": "Missing required field: title"  
}
```

Hinweise zur Fehlerbehandlung

- Bei fehlenden Pflichtfeldern (`title`, `status`, `url`) wird ein entsprechender Fehler zurückgegeben.
- Ungültige oder nicht vorhandene `id` bei GET, PUT oder DELETE resultieren in einer leeren Antwort oder einer Fehlermeldung.

Publisher-Verwaltung über die V6 REST-API

Einleitung

Die V6 REST-API ermöglicht es Admins, Publisher anzulegen und deren Details auszulesen. Die Funktionalität beschränkt sich aktuell auf **CREATE** und **READ**-Operationen.

Endpoint

Für Admins: `https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher`

CRUD-Operationen

POST: Publisher erstellen (CREATE)

Erstellt einen neuen Publisher und fügt die entsprechenden Daten in die Tabellen `global.logins` und `global.login_settings` ein.

Erforderliche Parameter:

Parameter	Typ	Beschreibung
<code>email</code>	String	Pflichtfeld, max. 255 Zeichen
<code>salutation</code>	String	Pflichtfeld, Werte: <code>mr</code> oder <code>mrs</code>
<code>prename</code>	String	Pflichtfeld, max. 255 Zeichen
<code>surname</code>	String	Pflichtfeld, max. 255 Zeichen

Optionale Parameter (Auszug): `company`, `street`, `zip`, `city`, `telephone`, `country` (ISO 3166 ALPHA-3), `billing_sepa_iban`, `companytype` (`un` oder `priv`), `language_interface` (Default: `DEU`), etc.

Beispiel-Request:

```
curl -X POST -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token:
ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" -d '{
  "email": "max.mustermann@example.com",
  "salutation": "mr",
  "prename": "Max",
  "surname": "Mustermann",
  "company": "Musterfirma",
  "street": "Musterstraße 1",
  "zip": "12345",
  "city": "Musterstadt",
  "country": "DEU",
  "billing_sepa_owner": "Max Mustermann",
  "billing_sepa_iban": "DE12345678901234567890",
  "billing_sepa_bic": "GENODEF1M01",
  "language_interface": "DEU",
  "billing_mode": 1,
  "billing_media": 1,
  "country_billing": "DEU",
  "country_publisher": "DEU",
  "billing_limit": 50
}' "https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher"
```

Erwartete Antwort:

```
{
  "id": 123,
  "name": "Max Mustermann",
  "email": "max.mustermann@example.com"
}
```

GET: Publisher auslesen (READ)

Gibt die Details eines spezifischen Publishers oder eine Liste von Publishern zurück.

Filteroptionen: Alle Felder der Tabelle können als Filter verwendet werden (z. B. `email`, `status`, `prename`, `surname`).

Beispiel-Request (spezifische ID):

```
curl -X GET -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token:
ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID"
"https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher?id=123"
```

Beispiel für GET mit Filtern:

```
curl -X GET -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token:
ADMIN_APIUSER_TOKEN" -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID"
"https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher?status=1&prename=Max"
```

Fehlerbehandlung

- **Fehlende Pflichtfelder (POST):**

```
{ "error": "Missing required field: email" }
```

- **Publisher nicht gefunden (GET):**

```
{ "error": "Publisher not found" }
```