

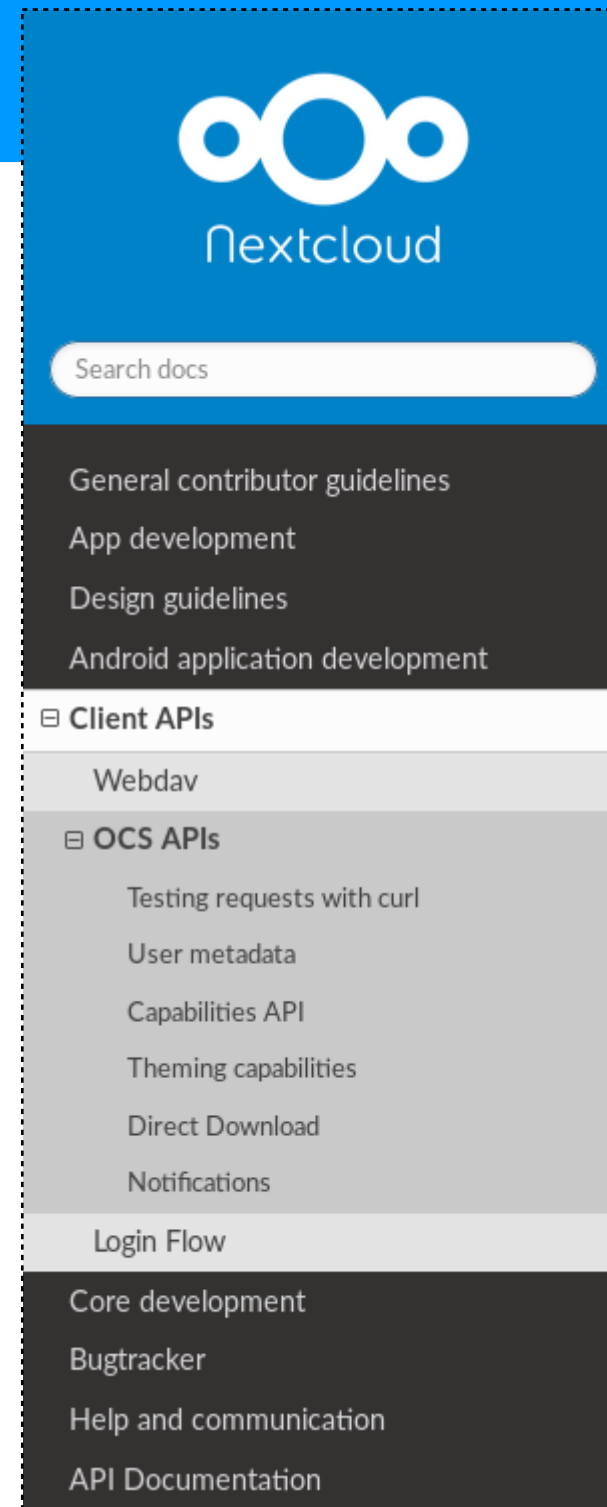
Verwendung der Nextcloud OCS API



Andreas Grupp
andreas@grupp-web.de

- Nextcloud-Administration über das Webinterface möglich, ...
 - aber im täglichen Handling aufwändig, langsam
- Wunsch nach einer Schnittstelle um per Skript / Anwendung z.B. auf Admin-Bereich zuzugreifen, um ...
 - User aus CSV-Datei heraus anzulegen
 - Gruppen zuzuordnen
 - Listen aller Art abzufragen
 - ...

- **Open Collaboration Services (OCS) – REST-basierte API**
 - Ursprung bei openDesktop.org als Teil des Social Desktop ... und man stößt dabei auf Frank Karlitschek
 - KDE, ownCloud, später Nextcloud, ...
- **REST → Representational State Transfer.**
 - Vereinfacht: Zustandslose Client-Server Programmierschnittstelle
 - Typisch für C/S-Webanwendungen mit http



Testing requests with curl

All OCS requests can be easily tested out using `curl` by specifying the request method (`GET`, `PUT`, etc) and setting a request body where needed.

For example: you can perform a `GET` request to get information about a user:

```
curl -u username:password -X GET 'https://cloud.example.com/ocs/v1.php/...' -H "OCS-APIRequest: true"
```

Obiger Beispiel-Code (vollständig, einzeilig):

```
curl -u username:password \  
-X GET 'https://cloud.example.com/ocs/v1.php/...' \  
-H "OCS-APIRequest: true"
```

Zweite Info im Admin-Manual

https://docs.nextcloud.com/server/16/admin_manual/configuration_user/user_provisioning_api.html

The base URL for all calls to the share API is `https://cloud.example.com/ocs/v1.php/cloud`.

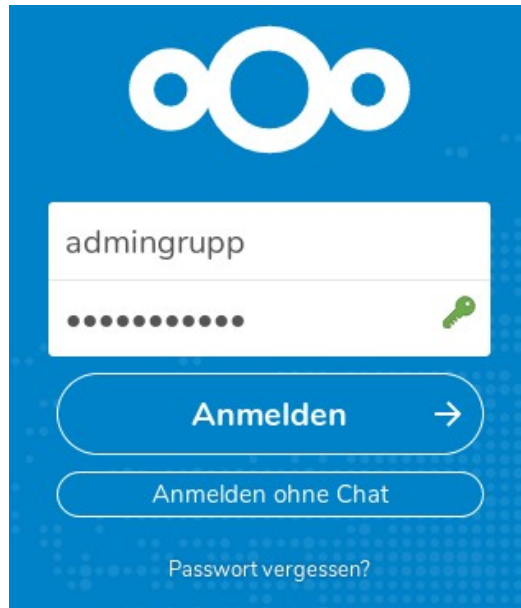
All calls to OCS endpoints require the `OCS-APIRequest` header to be set to `true`.

All POST requests require the `Content-Type: application/x-www-form-urlencoded` header. (Note: Some libraries like cURL set this header automatically, others require setting the header explicitly.)

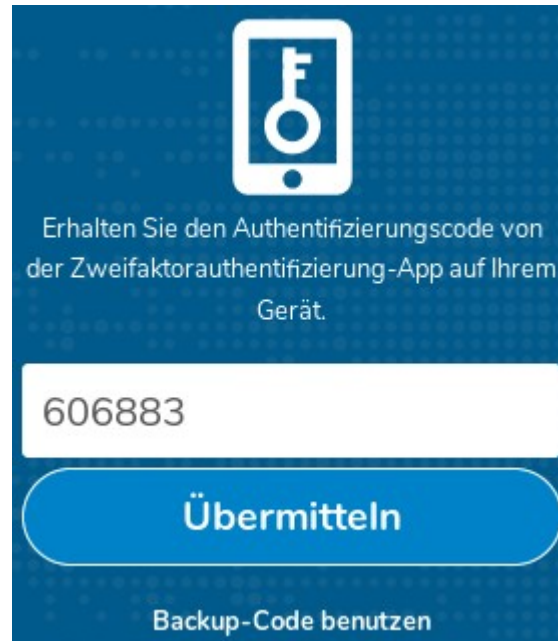
Setzen wir die Informationen zusammen:

- In Basis-URL eigene Nextcloud-URL einsetzen.
In meinem Fall ist das **`https://grupp-web.de/ownc`**
- Für einen Share-API-Zugriff ist **`/ocs/v1.php/cloud`** anfügen.
Gesamt in meinem Fall: **`https://grupp-web.de/ownc/ocs/v1.php/cloud`**
- Außerdem werden Zugangsdaten (**Benutzernamen** und **Passwort**) benötigt.
Auf keinen Fall die Web-Login-Daten verwenden!
App-Passwort erstellen!
Ist bei aktivierter 2-Faktor-Authentifizierung ohnehin erforderlich, da der API-Zugriff, wie auch andere Remote-Apps, nur Usernamen & Passwort kennen.

Nextcloud mit 2-FA und Erstellung App-Passwort



Nextcloud login screen. The logo is at the top. Below it is a text input field containing 'admingrupp' and a password field with dots. A green key icon is to the right of the password field. Below the fields are two buttons: 'Anmelden' with a right arrow and 'Anmelden ohne Chat'. At the bottom is a link 'Passwort vergessen?'.

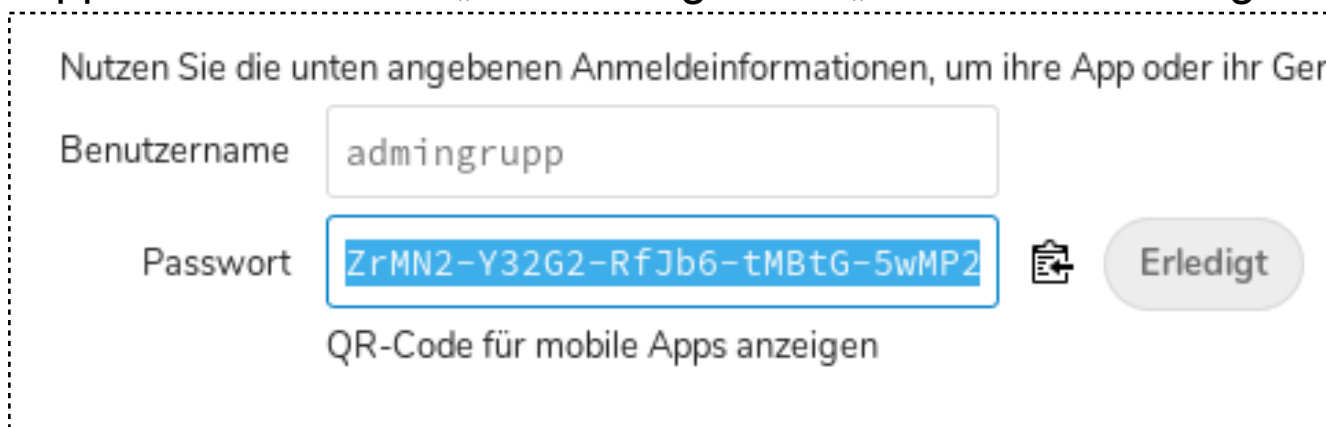


Nextcloud 2FA screen. At the top is an icon of a smartphone with a key. Below it is the text: 'Erhalten Sie den Authentifizierungscode von der Zweifaktorauthentifizierung-App auf Ihrem Gerät.' Below this is a text input field containing '606883'. Below the field is a large blue button 'Übermitteln'. At the bottom is a link 'Backup-Code benutzen'.

TOTP-Erzeugung im Normalfall auf einem 2. Gerät, z.B. LineageOS-Smartphone mit andOTP-App.

Zum TOTP-Verfahren gehören Key aber auch in so etwas wie KeePassXC speichern (kann übrigens auch TOTP – ist dann aber kein 2. Gerät)

App-Passwort unter „Einstellungen“ → „Sicherheit“ erzeugen:



Nextcloud App Password generation screen. At the top is the text: 'Nutzen Sie die unten angegebenen Anmeldeinformationen, um ihre App oder ihr Ger'. Below it are two fields: 'Benutzername' with 'admingrupp' and 'Passwort' with 'ZrMN2-Y32G2-RfJb6-tMBtG-5wMP2'. To the right of the password field is a clipboard icon and a button 'Erledigt'. Below the password field is the text 'QR-Code für mobile Apps anzeigen'.

Dritte Info in konkreter API-Doku für User-Management

https://docs.nextcloud.com/server/16/admin_manual/configuration_user/instruction_set_for_users.html

☐ User provisioning API

☐ Instruction set for users

Add a new user

Search/get users

Get data of a single user

Edit data of a single user

Disable a user

Enable a user

Delete a user

Get user's groups

Add user to group

Remove user from group

Promote user to subadmin

Demote user from subadmin

Get user's subadmin groups

Resend the welcome email

Instruction set for groups

Get data of a single user

Retrieves information about a single user. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: `ocs/v1.php/cloud/users/{userid}`

- HTTP method: GET

Status codes:

- 100 - successful

Example

```
$ curl -X GET http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank
```

- Returns information on the user Frank

Hier statt curl-Parameter „-u“ durch
Integration der Login-Daten in URL

Und das alles dann mal als Shell-Code zusammen:

Beispiel-Code 1 war:

```
curl -u username:password \  
      -X GET 'https://cloud.example.com/ocs/v1.php/...' \  
      -H "OCS-APIRequest: true"
```

API-Doku beschreibt u.a.:

Syntax: ocs/v1.php/cloud/users/{userid}

```
$ curl -X GET http://admin:secret@example.com/ocs/v1.php/cloud/users/  
Frank
```

Konkrete Umsetzungsmöglichkeit als Shell-Code:

```
USER=admingrupp  
PASS=ZrMN2-Y32G2-RfJb6-tMBtG-5wMP2  
URL=https://grupp-web.de/ownc/ocs  
curl -u ${USER}:${PASS} \  
      -X GET $URL'/ocs/v1.php/cloud/users/andreas' \  
      -H "OCS-APIRequest: true"
```


Ergebnis des vorigen curl-Aufrufs im XML-Format:


```
<?xml version="1.0"?>
<ocs>
<meta>
  <status>ok</status>
  <statuscode>100</statuscode>
  <message>OK</message>
  <totalitems></totalitems>
  <itemsperpage></itemsperpage>
</meta>
<data>
  <enabled>1</enabled>
  <storageLocation>/var/www/grupp-web.de/ocdata/andreas</
storageLocation>
  <id>andreas</id>
  <lastLogin>1561725202000</lastLogin>
  <backend>Database</backend>
  <subadmin/>
  <quota>
    <free>253982969856</free>
    <used>10519605883</used>
    <total>264502575739</total>
    <relative>3.98</relative>
    <quota>-3</quota>
  </quota>
```

```
<email>Andreas.Grupp@gmx.de</email>
<displayname>Andreas Grupp</displayname>
<phone>+49 7542 6462</phone>
<address>Quellenhalde 1, 88069 Tett nang</address>
<website>https://grupp-web.de/cms/</website>
<twitter></twitter>
<groups>
  <element>Fam. Grupp allg.</element>
  <element>Familie Irene und Andreas Grupp</element>
  <element>Freunde</element>
</groups>
<language>de_DE</language>
<locale>de_DE</locale>
<backendCapabilities>
  <setDisplayName>1</setDisplayName>
  <setPassword>1</setPassword>
</backendCapabilities>
</data>
</ocs>
```

Ergebnis des curl-Aufrufs auch als JSON-Data möglich:

curl Shell-Code für JSON-Data Ergebnis:

```
USER=admingrupp
PASS=ZrMN2-Y32G2-RfJb6-tMBtG-5wMP2
URL=https://grupp-web.de/ownc/ocs
curl -u ${USER}:${PASS} \
      -X GET $URL'/ocs/v1.php/cloud/users/andreas?format=json' \
      -H "OCS-APIRequest: true"
```



```
{"ocs":{"meta":
{"status":"ok","statuscode":100,"message":"OK","totalitems":"","items
perpage":""},"data":{"enabled":true,"storageLocation":"\\var\\www\\
grupp-web.de\\ocdata\\
andreas","id":"andreas","lastLogin":1561725202000,"backend":"Database
","subadmin":[],"quota":
{"free":253982367744,"used":10519605883,"total":264501973627,"relativ
e":3.98,"quota":-
3},"email":"Andreas.Grupp@gmx.de","displayname":"Andreas
Grupp","phone":"+49 7542 6462","address":"Quellenhalde 1, 88069
Tett nang","website":"https:\\/\\/grupp-web.de\\/cms\\/","twitter":"","gro
ups":["Fam. Grupp allg.,"Familie Irene und Andreas
Grupp","Freunde"],"language":"de_DE","locale":"de_DE","backendCapabil
ities":{"setDisplayname":true,"setPassword":true}}}}
```

Natürlich auch in anderen Sprachen möglich ...

```
#!/usr/bin/perl
#
use strict;
use DBI;
use Getopt::Long;
use MIME::Lite;
use XML::LibXML;

# ----- Konfiguration des Skripts (Start) -----
my $aduser      = 'admingrupp';
my $adpass      = 'ZrMN2-Y32G2-RfJb6-tMBtG-5wMP2';
my $ncUrl       = 'https://' . $aduser . ':' . $adpass . '@grupp-web.de/ownc/ocs/v1.php/cloud';
my $debug       = 0;    # 0=Kein Debugging, 1=Debugging
# ----- Konfiguration des Skripts (Ende) -----

binmode(STDOUT, ":utf8");

# ----- main() -----

sub main(){

    my ($allUserDom);

    if ($debug){ debug("Nextcloud-API-Base-URL: $ncUrl"); }

    $allUserDom = getAllUsers();
    foreach my $ncUserId ($allUserDom->findnodes('/ocs/data/users/element')) {
        if($debug){ print "main(): Abrufen der Daten für User-ID: " . $ncUserId->to_literal() . "\n"; }
        my $singleUserDom = getSingleUserData( $ncUserId->to_literal() );
        print "' " . $ncUserId->to_literal() . ", " .
            $singleUserDom->findnodes('/ocs/data/displayname') . ", " .
            $singleUserDom->findnodes('/ocs/data/email') . ", ";
        print "' ";
        my $komma = 0;
        foreach my $group ($singleUserDom->findnodes('/ocs/data/groups/element')){
            if($komma){ print ", "; }
            print $group->to_literal;
            $komma = 1;
        }
        print "' ";
        print "\n";
    }
}
```

... listet hier die User als CSV-Datei.

```
[agrupp@wwwneu nextcloud]$ ./list-nc-users-new-grupp-web
Rufe die Liste aller User ab ...
"admingrupp","Andreas Grupp [AdAcc]",root@grupp-web.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp,admin"
"andreas","Andreas Grupp",Andreas.Grupp@gmx.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp,Freunde"
"bernadette.grupp","Bernadette Grupp",b.grupp@gmx.de,"Fam. Grupp allg."
"clara","Clara Grupp",Clara.Grupp@gmx.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp"
"danny.petry","Danny Petry",danny.petry@gmx.de,""
"corbinian","Corbinian",corbinian.grupp@gmx.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp"
"marcel.grupp","Marcel Grupp",m.grupp@gmx.de,"Fam. Grupp allg."
"victor.liese","Victor Liese",vliese@denet.net.de,"Freunde"
"marcel.grupp","Marcel Grupp",marcel.grupp@grupp-11.de,"Fam. Grupp allg."
"elizabeth.grupp","Elizabeth Grupp",elizabeth.grupp@gmx.de,"Fam. Grupp allg."
"irene","Irene Grupp",Irene.Grupp@gmx.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp"
"iua.grupp","Irene & Andreas Grupp",iua.grupp@gmx.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp"
"jan.nathan","Jan Nathan",jan.nathan@gmx.de,"Freunde"
"joehannes.lehner","Johannes Eugen Lehner",joe@jehannes.de,""
"laura.grupp","Laura Grupp",Laura.Grupp@grupp11.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp"
"martha","Martha Grupp",Martha.Grupp@gmx.de,"Fam. Grupp allg.,Familie Irene und Andreas Grupp"
"muelersmann","Hans Muelersmann",andreas@grupp-web.de,""
"vera.schaeffer","Vera Schaeffer",vera.schaeffer@t-online.de,"Freunde"
```

Vielen Dank :-)

Kontaktdaten:

mailto:andreas@grupp-web.de

GnuPG-Key: 5FC0 2D54 A817 AE7F

CAcert-S/MIME-Zertifikat auf Anfrage verfügbar

xmpp:andreas@grupp-web.de

Web: <https://grupp-web.de/cms>

Mastodon: <https://social.tchncs.de/@angry>