

# Toon Open API documentation

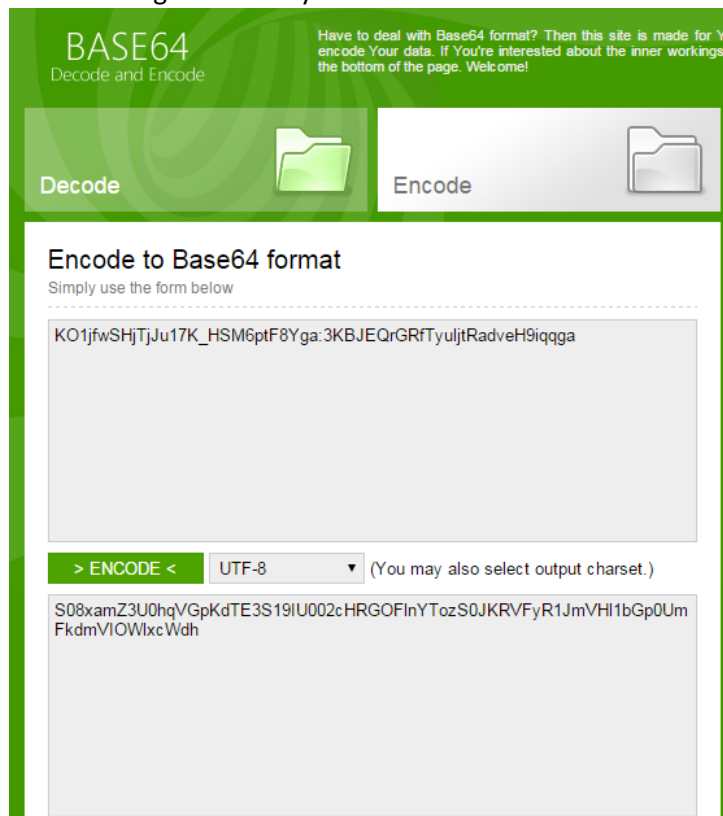
Version 1.1

## 1. Steps for credentials

- 1.1. Register on <http://www.toonapi.com/>. You will receive an email.
- 1.2. Verify your account by clicking the link in the received email. You will receive an email with the API Key, generated Password, Consumer-key and Consumer-secret.

## 2. Steps for using Node.js app

- 2.1. Go to <http://www.toonapi.com/documentation> and download the Toon API zip file.
- 2.2. Go to <https://nodejs.org/> and download and install Node.js.
- 2.3. Extract the zip file and edit the app.js.
- 2.4. Edit line 37 to contain your 'username' and line 38 to contain your 'password'. These are both literal and in between single quotes.
- 2.5. Open a command line windows and navigate to or stay in the folder where you have your app.js file. Run the command '`npm install`'.  
*For windows users: Hold Shift and right-click in the folder and then select 'Open command window here' to open a command line terminal.*
- 2.6. Generate your authentication header in one of the two ways described below.
- 2.6.1. Go to <https://www.base64encode.org/> and input Consumer-Key and Consumer-Secret in the following format. Key:secret



The screenshot shows the 'Base64 Encode and Decode' website. The 'Encode' tab is selected. The input field contains the text: `KO1jfwSHjTjJu17K_HSM6ptF8Yga:3KBJEQrGRfTyuljtRadveH9iqqga`. Below the input field, there is a green button labeled '> ENCODE <' and a dropdown menu set to 'UTF-8'. The output field displays the encoded result: `S08xamZ3U0hqVGpKdTE3S19IU002cHRGOFInYT0zS0JKRVFyR1JmVHl1bGp0UmFkdmVlOWlxcWdh`.

- 2.6.2. Go to <https://api.toonapi.com/store> and go to My Subscriptions. Select the pre-created ToonAPIApplication. Select the Curl command for GrantType: Password. Copy the

## Authentication Header in the curl command.

Test Show Keys

Keys - Production ▾

Consumer Key : KO1fwSHjTjJu17K\_HSM6ptF8Yga

Consumer Secret : 3KBJEQrGRfTyuljtRadveH9iqqga

Access Token : f250bb6f7b70cc06f3f84a8462aec1f3

cURL ⓘ

Validity Time : 3600 Seconds ⓘ Re-generate

```
curl -k -d "grant_type=password&username=<USER>&password=<PASSWORD>" -H "Authorization: Basic S08xamZ3U0hqVGpKdTE3S19lU002cHRGOFInYT0zS0JKRVFyR1JmVHl1bGp0UmFkdMVIOWlxcWdh" https://api.toonapi.com/token
```

Allowed Domains

ALL

The domains from which requests are allowed to the APIs. Leave empty or enter "ALL" to allow all domains.

Update Domains ⓘ

2.7. Edit line 52 in the app.js to contain your authentication header. The line will look like this:

**'Authorization': 'Basic YOUR\_AUTHENTICATION\_HEADER'**

2.8. Open a command line windows and navigate to or stay in the folder where you have your app.js file. Open the application in Node.js by entering the command 'node app.js' or 'npm start'.

*The application uses port 80 so you need to make sure this port is available. Skype for instance uses this port so you might need to shut this down. Also make sure your user has root/administrator rights*

2.9. Now that the application is running, you can go to <http://localhost:80> and press the Login button. You are now in the Toon API Simulator.

## 3. Steps for making REST calls to the API

3.1. Pick your weapon of choice for making REST calls. For instance cURL or Postman for Google Chrome.

3.2. Generate your authentication header in one of the two ways described below.

3.2.1. Go to <https://www.base64encode.org/> and input Consumer-Key and Consumer-Secret in the following format. Key:secret

BASE64  
Decode and Encode

Have to deal with Base64 format? Then this site is made for You. encode Your data. If You're interested about the inner workings of the bottom of the page. Welcome!

Decode Encode

Encode to Base64 format

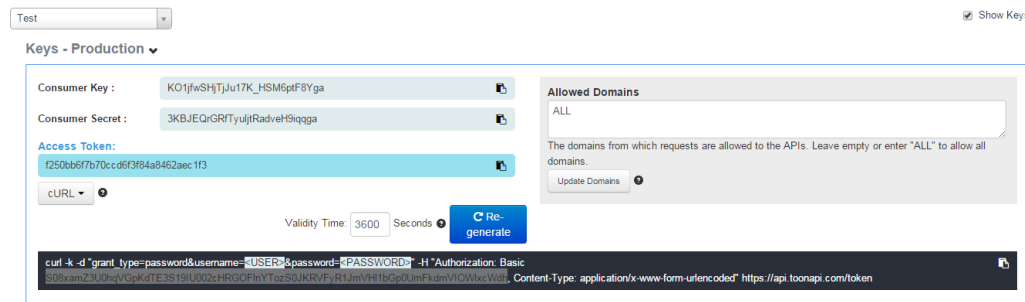
Simply use the form below

KO1fwSHjTjJu17K\_HSM6ptF8Yga:3KBJEQrGRfTyuljtRadveH9iqqga

> ENCODE < UTF-8 (You may also select output charset.)

S08xamZ3U0hqVGpKdTE3S19lU002cHRGOFInYT0zS0JKRVFyR1JmVHl1bGp0UmFkdMVIOWlxcWdh

- 3.2.2. Go to <https://api.toonapi.com/store> and go to My Subscriptions. Select the pre-created ToonAPIApplication. Select the Curl command for GrantType: Password. Copy the Authentication Header in the curl command.



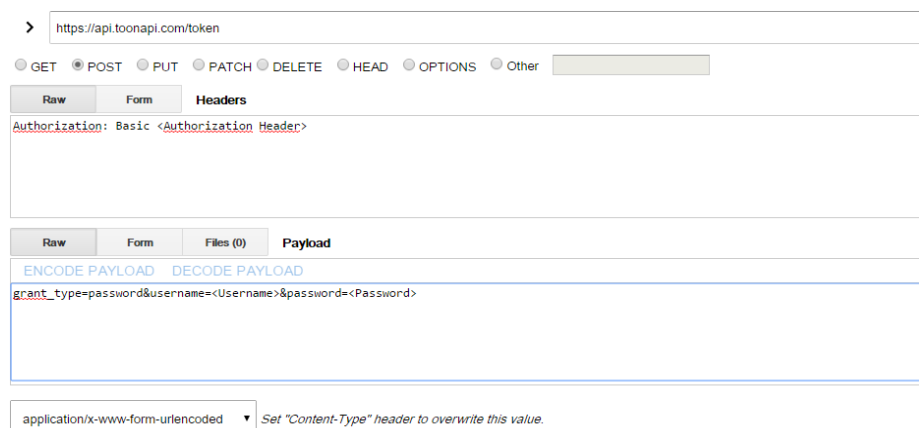
- 3.3. Make the call to the token endpoint to receive an access token. You can either import the whole cURL command from step 2b into a command line and replace the username and password field or create the call yourself.

- 3.4. The call comprises of a few elements. The url is <https://api.toonapi.com/token> and the method is POST.

Add two headers: Authorization: Basic **YOUR\_AUTHENTICATION\_HEADER**, Content-Type: x-www-form-urlencoded.

Add three elements to the body: grant\_type=password, username=**USERNAME**, password=**PASSWORD**.

You will receive a JSON object containing the access token.



*Example of the call in the Advaned Rest Client (Google Chrome Extension)*

- 3.5. All other calls to the API will have url <https://api.toonapi.com/toon/api/v1/OPERATION> and a header with the Access token. The next call you should execute is a GET call on /agreements. Add a header for Authorization: Bearer **ACCESS\_TOKEN**. You will receive an array of JSON objects with agreementIds. These agreements indicate to which Toon devices you have access.

- 3.6. You then need to set the Toon device you want to execute the next calls on. For example, if you want to change or retrieve the temperature for a certain Toon device, you need to set its agreementId as the current.

To set the agreementId of a certain Toon device make a POST call on /agreements. Add two headers: the standard authentication Bearer header as described in step 3.4. and Content-Type: application/json. Then add a body with the agreement you want to use. The JSON object can be copied from the previous call or created manually. An example can be seen below.

```
{
  "agreementId": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "street": "xxxxxxxxxxxxxxxxxxxxxxxx",
  "houseNumber": "xxx",
  "postalCode": "xxxxxx",
  "city": "xxxxxxxx",
  "displayCommonName": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "displaySoftwareVersion": "xxxxxxx"
}
```

Only the agreementId is required, the other fields are optional. If the agreement is accepted it returns a 200 OK Http status code.

- 3.7. You can now check the status of that Toon device with a GET call on /status. Add the default authorization header. This returns a JSON object with information on the Toon device like current temperature and program.