

## Machine-2-Machine Application Token

The application token is a 64-bit encrypted string that can be used in the M2M API 'login-token' endpoint to authenticate with this token instead of your ERS password. Each user is allowed up to 10 of these tokens, which can be revoked at any time.

**Note:** *The 'login' endpoint (used for authentication with the API) will be deprecated February 26, 2025. Application tokens are not impacted by password expiration or profile update requirements. It secures your credentials by providing the token in your code instead of password.*

### 1. Generate/View/Revoke Application Token

- a. Generate an application token
  - Navigate to ERS: <https://ers.cr.usgs.gov/> and login to your ERS account
  - Click "Create Application Token" under the "Application Tokens" section of the profile page

Application Tokens

Each user may create up to 10 tokens - these tokens can be used in the M2M API 'login-token' endpoint to authenticate with this token instead of your ERS password.

Create Application Token

Token Description	Token Scope	Creation Date	Expire Date	Last Login Date	Revoke
No application tokens have been created					

- On the Generate Application Token page, you can customize the token name by entering text into the "Token Name" text field. If you leave the token name empty, it will be generated automatically by the system.
- Set "Token Expiration" to set expiration on a specific date (not required - if not provided, the token does not expire)
- Choose an Application in the Allowed Applications / Token Scope table. Token scopes allow you to create a token that is only valid for the applications selected. If this token were lost or compromised, only the allowed scopes (applications) would be affected. We recommend only allowing as many scopes as your token requires access to work. For instance, if your script is only accessing M2M, allowing only the 'M2M API' scope is recommended.
- Click the "Generate Application Token" button.

## Generate Application Token

Application tokens can be used in the M2M API 'login-token' endpoint to authenticate with this token instead of your ERS password. Each user is allowed up to 10 of these tokens, which can be revoked at any time. Tokens do not automatically expire unless given an expiration time at the time of creation.

To generate an application token, select at least one scope and then click the button below.

**You will have 60 seconds to record your token before it is cleared from the screen.**

In the event that you do not record your token within the allotted time, revoke the new token and generate a new one.

Token Name	(Leave empty for auto-generated name)
Token Expiration	mm / dd / yyyy <input type="button" value="Calendar"/>
<b>Allowed Applications / Token Scope</b>	
<input type="checkbox"/>	M2M API
<b>At least one selection is required</b>	
	<input type="button" value="Generate Application Token"/>

- Record the token value in a safe place. Protect this token as you do your password. If the token is compromised or is no longer needed for a project, you may revoke the token (the instructions can be found in section c. Revoke Application Token) without affecting any other settings on your ERS account. The token will be cleared from the screen after 60 seconds and cannot be recovered or displayed after that time. If you don't record your token within the allotted time, revoke this token and generate a new one.

## Generate Application Token

Application tokens can be used in the M2M API 'login-token' endpoint to authenticate with this token instead of the your ERS password. Each user is allowed up to 10 of these tokens, which can be revoked at any time. Tokens do not automatically expire unless given an expiration time at the time of creation.

To generate an application token, select at least one scope and then click the button below.

**You will have 60 seconds to record your token before it is cleared from the screen.**

In the event that you do not record your token within the allotted time, revoke the new token and generate a new one.

Token Value

gi4VfQHC4I9Xdncxr8GR@TmPboGx5wADiKlmQ3oej5uN7wtpZZsHrIJgsFik1TM1

Token will be cleared from the screen in 35 seconds

Token Name (Leave empty for auto-generated name)

Token Expiration mm / dd / yyyy

Token scopes allow you to create a token that is only valid for the applications selected. If this token were lost or compromised, only the allowed scopes (applications) would be affected. We recommend only allowing as many scopes as your token requires access to in order to work. For instance, if your script is only accessing M2M, allowing only the 'M2M API' scope is recommended.

### Allowed Applications / Token Scope

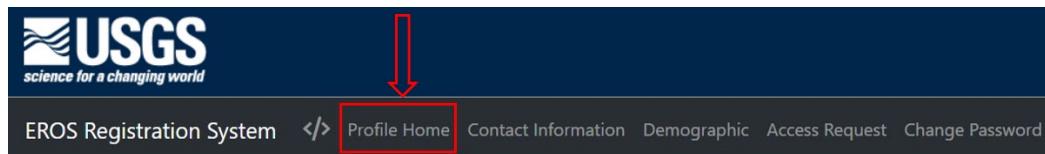
M2M API

At least one selection is required

Token Generated

### b. View Application Token

- The application tokens associated with your account can be found on the profile home page. Login to ERS and navigate to the "Profile Home" page



- A list of application tokens is shown up in the "Application Tokens" section. Each application token has the token description (token name), creation date, and last login date through M2M. Un-attempted tokens are marked 'unused'.

### Application Tokens

Each user may create up to 10 tokens - these tokens can be used in the M2M API 'login-token' endpoint to authenticate with this token instead of the your ERS password.

Create Application Token

Token Description	Token Scope	Creation Date	Expire Date	Last Login Date	Revoke
wf2LaKWgxtrUMX6g	• M2M API	2024-07-17T14:00:15-05:00	(does not expire)	(unused)	X
ME6Zl9p1c01Fq7cM	• M2M API	2024-07-17T14:01:29-05:00	(does not expire)	(unused)	X

### c. Revoke Application Token

Each user can have up to 10 application tokens. If you reach the token limit, you won't be able to create application tokens unless you revoke an existing one. The existing application tokens can be revoked at any time. To revoke an application token:

- Login to ERS and navigate to the profile page by clicking "Profile Home" at the top of the menu
- Click the 'x' icon next to the application token that you want to revoke. The token will be removed from your account and can't be used to authenticate through M2M anymore.

Token Description	Token Scope	Creation Date	Expire Date	Last Login Date	Revoke
wf2LaKWgxtrUMX6g	• M2M API	2024-07-17T14:00:15-05:00	(does not expire)	(unused)	X
ME6Zk9p1c01Fq7cM	• M2M API	2024-07-17T14:01:29-05:00	(does not expire)	(unused)	X

## 2. Use Application Token

The application tokens can be used in the M2M API endpoint “login-token” to generate an API key without a password. You can get the API key by the M2M test page or your scripts.

### a. M2M Test Page

- Navigate to the M2M test page: <https://m2m.cr.usgs.gov/api/test/json/> and select the “login-token” endpoint

Service Endpoints

login-token

- Enter your ERS username in the “username” field, and recall the created token value (Token creation can be found in 1.a Generate an Application Token) and use it in the “token” text field

Method Details				
Endpoint: login-token				
Input Parameters				
Value	Parameter Name	Required?	Data Type	Description
my-username	username	Yes	string	ERS Username
OWN8a5Gp4DHAKaahUIWaefAV!m@bCbUvsE	token	Yes	string	Application Token
Value	Parameter Name	Required?	Data Type	Description

Submit Request

- Click “Submit Request” and the “API key” is returned as ‘data’ in the response (You have to scroll to the most right to get the whole key)

Response			
Request ID	HTTP Status Code	Error Code	Error Message
14631893	200		
<pre>{   "requestId": 14631893,   "version": "stable",   "data": "eyJjaWQlOjQxOTc2NCwicyI6IjE2NzY0MDk4NDY1LC3yIjoyMTQsInA10lsidXNlcIisImFwcGxpY2F0al9uIiwiYnxsb3ctZGVidWc1LCJibWE1LC3kZHhlLC3kZXZlbg9wZXIifLC3k" }</pre>			

### b. Sample Script – PHP

```
<?php
try{
```

```

$serviceUrl = 'https://m2m.cr.usgs.gov/api/api/json/stable/';
$url = $serviceUrl.'login-token';

	payload = array(
		"username" => $my_user_name,
		"token" => $my_token_value
);

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'POST');
curl_setopt($ch, CURLOPT_USERAGENT, 'PHP v' . PHP_VERSION);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$output = curl_exec($ch);

try {
    $httpStatusCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    if ($output == null) {
        throw new Exception('No output from service');
    }

    $results = json_decode($output);
    if ($results === false) {
        throw new Exception('Could not parse JSON');
    }
    if ($results->errorCode != null) {
        print_r($results);
        throw new Exception("{$results->errorCode} - {$results->errorMessage}");
    }
    if ($httpStatusCode == 404) {
        throw new Exception('404 Not Found');
    } elseif($httpStatusCode == 401) {
        throw new Exception('401 Unauthorized');
    } elseif($httpStatusCode >= 400) {
        //500 errors mean there is an error but it may have been caught
        throw new Exception('Error Code: ' . $httpStatusCode);
    }
} catch (Exception $e) {
    curl_close($ch);
    throw new Exception("Error communicating with service ({$httpStatusCode}
- {$url}): {$e->getMessage()}");
}
curl_close($ch);
$apiKey = $results->data;
echo "API Key: {$apiKey}\n";
} catch (Exception $e) {
    die("Error: {$e->getMessage()}\n\n");
}

```

### c. Sample Script – Python

```
import json
```

072024

```
import requests
import sys

serviceUrl = "https://m2m.cr.usgs.gov/api/api/json/stable/"
url = serviceUrl + "login-token"
# login-token
payload = {"username" : "my_user_name", "token" : "my_token_value"}

json_data = json.dumps(payload)
response = requests.post(url, json_data)

try:
    httpStatusCode = response.status_code
    if response == None:
        print("No output from service")
        sys.exit()

    output = json.loads(response.text)
    if output['errorCode'] != None:
        print(output)
        sys.exit()
    if httpStatusCode == 404:
        print("404 Not Found")
        sys.exit()
    elif httpStatusCode == 403:
        print("401 Unauthorized")
        sys.exit()
    elif httpStatusCode == 500:
        print("Error Code", httpStatusCode)
        sys.exit()
except Exception as e:
    response.close()
    print(e)

response.close()
apiKey = output['data']

print("API Key: " + apiKey)
```

You can supply this API key in the HTTP Header 'X-Auth-Token' in the M2M API requests.