

Development Guide for  
**Synology SSO Server**



# Table of Contents

Chapter 1: Introduction	01
Chapter 2: Javascript SDK DSM JavaScript SDK Script Location Usage	02
Chapter 3: Manual Flow	05
Chapter 4: Exchange User Information Exchange for user's information	06
Chapter 5: Example Code Javascript SDK Examples	07
Chapter 6: Error Strings ERR_STRING	10

## Find your information

Synology publishes a wide range of supporting documentation.

In [Knowledge Center](#), you will find useful Help and FAQ articles, as well as video tutorials breaking up processes into handy steps. You can also find User's Guides, Solution Guides, brochures, and White Papers. Experienced users and administrators will find answers and guidance in technical Administrator's Guides and Developer Guides.

Got a problem and unable to find the solution in our official documentation? Search hundreds of answers by users and support staff in [Synology Community](#) or reach [Synology Support](#) through the web form, email or telephone.



# Chapter 1: Introduction

Synology DSM SSO Server is based on the OAuth 2 protocol. We provide the JavaScript SDK for 3rd party development. SSO Server JavaScript SDK script will be installed automatically after SSO Server installation.

# Chapter 2: Javascript SDK

## DSM JavaScript SDK Script Location

```
http://DSM_IP_OR_HOSTNAME:5000/webman/sso/synOSSO-1.0.0.js
```

## Usage

### Initialization

```
SYNOSSO.init
```

`SYNOSSO.init` is used to initialize `SYNOSSO` SDK. You need to call `SYNOSSO.init` before calling any other `SYNOSSO` APIs.

Function parameters of `SSOSYNO.init`:

Key	Value	Description
<code>oauthserver_url</code>	string	The URL of the DSM where SSO Server is installed.
<code>app_id</code>	string	APP ID registered on the DSM SSO Server
<code>redirect_uri</code>	string	Redirect URI registered on the DSM SSO Server.
<code>callback</code>	Javascript function object	User defined callback for handling login query/login response.
<code>domain_name(optional)</code>	string	Windows AD domain name of SSO client. Ex: "MYDOMAIN.COM"
<code>ldap_baseDN(optional)</code>	string	LDAP baseDN of SSO client. Ex: "dc=myldap,dc=com"

#### Notes:

- Directory service related options are for directory service checking. If one of these options is provided, SSO Server will validate if this directory service is the same as the DSM that the SSO Server belongs to.

**Example:**

```
SYNOSSO.init({
  oauthserver_url: 'http://10.13.20.131:5000',
  app_id: '153fcb35b01571b49cb0adca3a4bda40',
  redirect_uri: 'http://10.13.20.130:5000', //redirect url have to be
  the same as the one registered in SSO server, and can be a plain text
  html file.
  callback: authCallback
});
```

## Authentication

```
SYNOSSO.login();
```

After calling `SYNOSSO.login`, a login popup window containing a dialog for SSO will appear. `SYNOSSO.login` has no arguments and will call the callback registered in `SYNOSSO.init` after the user logs in successfully.

**Example:**

```
SYNOSSO.login();
```

**Response:**

Response of callback registered in `SYNOSSO.init()`:

Key	Value	Description
Status	String: "login"/"not_login"/ ERR_STRING	Show status of this user on SSO Server.
Access_token	string	Access token returned from SSO Server after this user logs in successfully.

**If the user already signed in to SSO Server:**

```
response:{
  status: 'login',
  access_token: 'ABCDE'
}
```

**If the user hasn't signed in to SSO Server:**

```
response:{
  status: 'not_login'
}
```

**If any unexpected error occurs:**

```
response: {
  status: 'ERR_STRING'
}
```

**Notes:**

- For ERR\_STRING, please refer to Chapter 6 for more details.

**Logout**

```
SYNOSSO.logout(function() {
  //do something after logout.
});
```

**Function parameters of SSOSYNO.logout:**

Key	Value	Description
callback	Javascript function	The callback which will be called after the user logs out from SSO Server.

`SYNOSSO.logout` has a callback which will be called after the user signs out of SSO Server.

- Before a user signs out of your application, call `SYNOSSO.logout`, and it will sign out this user from SSO Server.
- `SYNOSSO.init` must be called before `SYNOSSO.logout`.
- `SYNOSSO.logout` only signs out the user from SSO Server and will not affect the login status of users in other applications.

Response of callback of `SYNOSSO.logout` has no arguments.

# Chapter 3: Manual Flow

1. Bring the user to `http://[DSM OAuth Server:5000]/webman/sso/SSOAuth.cgi` with the following query string parameters:

- **app\_id** : APP ID registered on DSM SSO Server
- **redirect\_uri** : Redirect URI registered on DSM SSO Server
- **synossoJSSDK**: False, represents the manual flow
- **scope**: SSO server only provides “user\_id” scope, which means there is limited user information for Single-Sign On
- **state** (optional) : Used to protect CSRF

Then, the login window will appear for the user to input their username and password.

## Example:

```
SSO Server: 10.13.20.254
SSO Client: 10.13.22.128
http://10.13.20.254:5000/webman/sso/SSOAuth.cgi?app_
id=a5a78d55b7d30dab1b3067d26bc4_9e49&scope=user_
id&synossoJSSDK=false&redirect_uri=http://10.13.22.128:5000
```

2. User signs in to SSO Server.

3. After signing in successfully, the user will be redirected back to the redirect URI that this app registered on SSO Server with following hash values:

- **access\_token**: The access token which will be used to exchange user information.
- **State** (optional): If you provide the state in step 1, the exact same state will be returned.

## Example:

```
http://10.13.22.128:5000#access_token=58322f3eaaG7t69030edH2bcdee08brWc6
250eba&state=fabc21cf
```

# Chapter 4: Exchange User Information

## Exchange for user's information

1. You need to use an access token to get the user\_id and user\_name.
2. Go to endpoint: `http://[DSMOauthServer:5000]/webman/sso/SSOAccessToken.cgi` with these query string parameters:
  - action: "exchange"
  - access\_token: "ABCDE"
  - app\_id: "asfsf sdfsdf3e"

### Example:

```
curl
http://[DSMOauthServer:5000]/webman/sso/SSOAccessToken.
cgi?action="exchange"&access_token="ABCDE"&app_id="asfsfsdfsdf3e"
```

### Response:

If the token is correct:

```
{
  success: true,
  data: {
    user_id: 1024,
    user_name: john
  }
}
```

If any unexpected errors occurred:

```
{
  success: false,
  error: 'ERR_STRING'
}
```



# Chapter 5: Example Code

## Javascript SDK Examples

### Fontpage.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Test App 1</title>
</head>
<body>
  <h1>Test App 1</h1>
  <h2>Please sign in via Synology SSO</h2>
  <button id="login-button">SSO Login</button>
  <h3 id="my-text"></h3>
</body>
<script type="text/javascript" src="https://sso.server/webman/sso/synoSSO-1.0.0.js"></script>
<script>
document.addEventListener("DOMContentLoaded", function() {
  SYNOSSEO.init({
    oauthserver_url: 'https://sso.server/',
    app_id: '34744cd97de19fec4272cb99d13aca75',
    redirect_uri: 'https://10.17.188.213/',
    callback: authCallback
  })
  function authCallback(response) {
    if('not_login' === response.status) { //user not
      console.log(response.status);
    }
  }
});
</script>
```

```

    } else if('login' === response.status) {
        var xhr = new XMLHttpRequest();
        xhr.open('GET', '/login_backend.php?accesstoken=' + response.
access_token);
        xhr.onload = function() {
            if (xhr.status === 200) {
                var message = "";
                var response = JSON.parse(xhr.responseText);
                if (response && response.success && response.data) {
                    message = 'Login success, User Name=' + response.data.user_
name + ', User ID=' + response.data.user_id;
                } else {
                    message = 'Login failed. Response: ' + xhr.responseText;
                }
                document.getElementById("my-text").innerHTML = message;
            } else {
                alert('Request failed. Returned status of ' + xhr.status);
            }
        };
        xhr.onerror = function() {
            alert('Request failed due to network error.');
```

## Login\_backend.php

```
<?php
$accesstoken = $_GET['accesstoken'];
function httpGet ($url)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false); //for testing ignore
checking CA
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    $output=curl_exec($ch);
    curl_close($ch);
    return $output;
}
$url_str = "https://sso.syno/webman/sso/SSOAccessToken.
cgi?action=exchange&access_token=".$accesstoken;
header('Content-Type: application/json');
echo httpGet($url_str);

?>
```

# Chapter 6: Error Strings

## **ERR\_STRING**

- `server_error` - SSO server error.
- `parameter_error` - Parameter error when `SYNOSSO.init`.
- `invalid_app_id` - `APP_ID` error.
- `invalid_redirect_uri` - Redirect URI error.
- `invalid_directory_service` - Different directory service between `SYNOSSO.init` and DSM SSO Server.
- `invalid_token` - Invalid SSO access token.
- `unknown_error` - Other unexpected errors.



**SYNOLOGY  
INC.**

9F, No. 1, Yuandong Rd.  
Banqiao Dist., New Taipei City 220545  
Taiwan  
Tel: +886 2 2955 1814

**SYNOLOGY  
AMERICA CORP.**

3535 Factoria Blvd SE, Suite #200,  
Bellevue, WA 98006  
USA  
Tel: +1 425 818 1587

**SYNOLOGY  
UK LTD.**

Unit 5 Danbury Court, Linford Wood,  
Milton Keynes, MK14 6PL  
United Kingdom  
Tel.: +44 (0)1908048029

**SYNOLOGY  
FRANCE**

102 Terrasse Boieldieu (TOUR W)  
92800 Puteaux  
France  
Tel: +33 147 176288

**SYNOLOGY  
GMBH**

Grafenberger Allee 295  
40237 Düsseldorf  
Deutschland  
Tel: +49 211 9666 9666

**SYNOLOGY  
SHANGHAI**

200070, Room 201,  
No. 511 Tianmu W. Rd.,  
Jingan Dist., Shanghai,  
China

**SYNOLOGY  
JAPAN CO., LTD.**

4F, No. 3-1-2, Higashikanda,  
Chiyoda-ku, Tokyo, 101-0031  
Japan

**Synology®**



[synology.com](https://synology.com)

Synology may make changes to specifications and product descriptions at any time, without notice. Copyright © 2023 Synology Inc. All rights reserved. ® Synology and other names of Synology Products are proprietary marks or registered trademarks of Synology Inc. Other products and company names mentioned herein are trademarks of their respective holders.