Table of Contents

Chapter 1: Introduction

Chapter 2: Usage

Initialization .................................................................................................................. 5
Configuring Options ..................................................................................................... 5
Authentication ............................................................................................................... 5
Logout ........................................................................................................................... 5
Callback of SYNOSSO.init .......................................................................................... 6
Synology DSM SSO Server is based on OAuth2 protocol. We provide the JavaScript SDK for 3rd party development. SSO Server JavaScript SDK script will be installed automatically after SSO Server installation.

**DSM JavaScript SDK script location**

http://DSM_IP_OR_HOSTNAME:5000/webman/sso/synoSSO-1.0.0.js
Usage

Initialization

```javascript
SYNOSSO.init({
    app_id: '153fcb35b01571b49cb0a5ca3a4bda40',
    redirect_uri: 'http://10.13.20.130/relay.html', // redirect uri have to be the same as the one registered in SSO server, and can be a plain text html file.
    callback: authCallback
});
```

Configuring Options

Required options

The following four options are required.

- **oauthserver_url**: The URL of DSM which installed SSO Server.
- **app_id**: APP ID registered on DSM SSO Server.
- **redirect_uri**: Redirect URI registered on DSM SSO Server.
- **callback**: User defined callback for handling login/query/login response.

Optional Options

These two directory service related options are used for directory service check. If any one of these options is provided, SSO Server will validate if this directory service is the same as the DSM on which SSO Server is operating.

- **domain_name** (optional): SSO client windows AD domain name.
  
  E.g., **domain_name**: "MYDOMAIN.COM"

- **ldap_baseDN** (optional): SSO client LDAP base DN
  
  E.g., **ldap_baseDN**: "dc=myldap,dc=com"

Authentication

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

This method has no arguments and will call the callback which is registered in `SYNOSSO.init` after a user logs in successfully. After calling `SYNOSSO.login`, a login window containing a dialog for SSO will pop out.

Logout

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

`SYNOSSO.logout` has a callback which will be called after user logout from SSO Server.

- Before a user logs out from your application, call `SYNOSSO.logout`, and this method will log out this user from SSO Server.
• SYNOSO.init must be called before SYNOSO.logout.
• SYNOSO.logout only logs out the user from SSO Server and will not affect login status of the user in others applications.

# Callback of SYNOSO.init

## Response

- If the user already login SSO Server:

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

- If the user didn’t login SSO Server:

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

- If any unexpected error occurred:

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

The variable **ERR_STRING** may be replaced by the error codes listed in the table below:

<table>
<thead>
<tr>
<th>No.</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>server_error</td>
<td>SSO server error.</td>
</tr>
<tr>
<td>2</td>
<td>parameter_error</td>
<td>Parameter error when SYNOSO.init.</td>
</tr>
<tr>
<td>3</td>
<td>invalid_app_id</td>
<td>APP_ID error.</td>
</tr>
<tr>
<td>4</td>
<td>invalid_redirect_uri</td>
<td>Redirect URI error.</td>
</tr>
<tr>
<td>5</td>
<td>invalid_directory_service</td>
<td>Different directory service between SYNOSO.init and DSM SSO Server.</td>
</tr>
<tr>
<td>6</td>
<td>invalid_token</td>
<td>Invalid SSO access token.</td>
</tr>
<tr>
<td>7</td>
<td>unknown_error</td>
<td>Other unexpected errors.</td>
</tr>
</tbody>
</table>