

# OAuthServiceBuilder



Apr 04, 2024 04:22

Supported Clients

NGClient

## Methods Summary

<code>OAuthServiceBuilder</code> <code>additionalParameters(params)</code>	Add some more parameters to the authorization url.
<code>OAuthService</code> <code>build(api)</code>	Creates an OAuth service that can be used to obtain an access token and access protected data.
<code>OAuthService</code> <code>build(api)</code>	Creates an OAuth service that can be used to obtain an access token and access protected data.
<code>OAuthServiceBuilder</code> <code>callback(callback, timeout)</code>	Configure the service with a callback function to be executed when the service is ready to use.
<code>OAuthServiceBuilder</code> <code>clientSecret(clientSecret)</code>	Set the client secret of the application.
<code>OAuthServiceBuilder</code> <code>deeplink(deeplink)</code>	OPTIONAL This is a way to override the default deeplink method name, which is 'deeplink_svy_oauth'.
<code>OAuthServiceBuilder</code> <code>defaultScope(scope)</code>	Request always the same scope.
<code>OAuthServiceBuilder</code> <code>domain(domain)</code>	Set the domain if the API supports it (e.
<code>String</code> <code>getUsedAuthorizationURL(api)</code>	Get the authorization url.
<code>String</code> <code>getUsedAuthorizationURL(api)</code>	Get the authorization url.
<code>OAuthServiceBuilder</code> <code>responseMode(mode)</code>	Configure if the code/tokens are going to be received as a query or as a url fragment.
<code>OAuthServiceBuilder</code> <code>responseType(response_type)</code>	Configures the OAuth flow.
<code>OAuthServiceBuilder</code> <code>scope(scope)</code>	Request any unique scope per each access token request.
<code>OAuthServiceBuilder</code> <code>state(state)</code>	Configures the anti forgery session state.
<code>OAuthServiceBuilder</code> <code>tenant(tenant)</code>	Set the tenant identifiers/organization if the API supports it (e.
<code>OAuthServiceBuilder</code> <code>withPKCE()</code>	Configures the service with Proof Key for Code Exchange, which prevents authorization code interception attacks.

## Methods Details

### additionalParameters(params)

Add some more parameters to the authorization url.

#### Parameters

`Object` `params` a json containing the parameters and their values e.g. `{'param1': 'value1', 'param2': 'value2'}`

#### Returns

`OAuthServiceBuilder` the service builder for method chaining

#### Supported Clients

NGClient

#### Sample

### build(api)

Creates an OAuth service that can be used to obtain an access token and access protected data.

#### Parameters

`CustomApiBuilder` `api` a custom api, see `plugins.oauth.customApi`

#### Returns

`OAuthService` an `OAuthService` object that can be used to make signed requests to the api

#### Supported Clients

NGClient

**Sample****build(api)**

Creates an OAuth service that can be used to obtain an access token and access protected data.

**Parameters**

[String](#) api an OAuth provider id, see `plugins.oauth.OAuthProviders`

**Returns**

[OAuthService](#) an OAuthService object that can be used to make signed requests to the api

**Supported Clients**

NGClient

**Sample****callback(callback, timeout)**

Configure the service with a callback function to be executed when the service is ready to use. After the access token is returned by the server, this callback function is executed.

**Parameters**

[Func](#) callback a function in a scope or form  
[ion](#) ack

[Num](#) timeout max number of seconds in which the callback method should be executed (with success or error message) Please note that the timeout  
[ber](#) ut should be enough for the user to login and accept permissions.

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

**Supported Clients**

NGClient

**Sample****clientSecret(clientSecret)**

Set the client secret of the application.

**Parameters**

[String](#) clientSecret a secret known only to the application and the authorization server

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

**Supported Clients**

NGClient

**Sample****deeplink(deeplink)**

OPTIONAL This is a way to override the default deeplink method name, which is 'deeplink\_svy\_oauth'. The deeplink method is a global method that receives the code needed to obtain the access token from the OAuth provider.

NOTE: The deeplink method name is strongly related to the redirect url configured for the application. If the OAuth provider (eg. Microsoft AD, LinkedIn) requires to configure a full redirect url then it should be of the form:  
[https://example.com/<solution\\_name>/m/<deeplinkmethod>](#) - where <deeplinkmethod> is the name configured with the service builder  
[https://example.com/<solution\\_name>/m/deeplink\\_svy\\_oauth](#) - if the deeplink method name was not overridden

If the deeplink method with the provided name does not exist in the solution, then a default deeplink method is generated under the hood with the solution model. If a global method with the provided name already exists in the solution, then it should set the access token on the service and handle possible errors.

**Parameters**

[String](#) deeplink a global scope method name

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

---

**Supported Clients**

NGClient

**Sample****defaultScope(scope)**

Request always the same scope.

Scope is a mechanism in OAuth 2.0 to limit an application's access to a user's account.

An application can request one or more scopes, separated by space.

This information is then presented to the user in the consent screen, and the access token issued to the application will be limited to the scopes granted.

**Parameters**

[String](#) scope the default scope

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

**Supported Clients**

NGClient

**Sample****domain(domain)**

Set the domain if the API supports it (e.g.Okta)

**Parameters**

[String](#) domain ;

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

**Supported Clients**

NGClient

**Sample****getUsedAuthorizationURL(api)**

Get the authorization url. This is for DEBUGGING PURPOSES ONLY.

**Parameters**

[CustomApiBuilder](#) api a custom api builder

**Returns**

[String](#) the used authorization url

**Supported Clients**

NGClient

**Sample****getUsedAuthorizationURL(api)**

Get the authorization url. This is for DEBUGGING PURPOSES ONLY.

**Parameters**

[String](#) api an OAuth provider id, see plugins.oauth.OAuthProviders

**Returns**

[String](#) the used authorization url

**Supported Clients**

NGClient

**Sample****responseMode(mode)**

---

Configure if the code/tokens are going to be received as a query or as a url fragment.  
Will be ignored if the response type is token/id\_token or if the oauth provider does not support it.

For the "fragment" response mode the redirect url configured for the oauth app needs to be of the following form  
https://example.com/servoy-service/oauth/solutions/<solution\_name>/m/<deeplinkmethod> - where <deeplinkmethod>  
is the name configured with the service builder

#### Parameters

[String](#) mode can be "query" or "fragment"

#### Returns

[OAuthServiceBuilder](#) the service builder for method chaining

#### Supported Clients

NGClient

#### Sample

### responseType(response\_type)

Configures the OAuth flow. Defaults to "code" (authorization code flow) if not set.  
Use response type "token" for the implicit grant flow.  
Use response type "id\_token" for OpenID Connect sign-in. In this case the response is a JWT token which can be used to verify the identity of a user.  
OAuth providers may allow combinations of "code" "id\_token" "token".

#### Parameters

[String](#) response\_type one or a combination of "code" "id\_token" "token"

#### Returns

[OAuthServiceBuilder](#) the service builder for method chaining

#### Supported Clients

NGClient

#### Sample

### scope(scope)

Request any unique scope per each access token request.  
Scope is a mechanism in OAuth 2.0 to limit an application's access to a user's account.  
An application can request one or more scopes, separated by space.  
This information is then presented to the user in the consent screen, and the access token issued to the application will be limited to the scopes granted.

#### Parameters

[String](#) scope one or multiple scopes separated by space

#### Returns

[OAuthServiceBuilder](#) the service builder for method chaining

#### Supported Clients

NGClient

#### Sample

### state(state)

Configures the anti forgery session state. This is required in some APIs (like Facebook's).

#### Parameters

[String](#) state ;

#### Returns

[OAuthServiceBuilder](#) the service builder for method chaining

#### Supported Clients

NGClient

#### Sample

### tenant(tenant)

Set the tenant identifiers/organization if the API supports it (e.g. Microsoft AD)

---

**Parameters**

[String](#) tenant ;

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

**Supported Clients**

NGClient

**Sample****withPKCE()**

Configures the service with Proof Key for Code Exchange, which prevents authorization code interception attacks. See more at <https://datatracker.ietf.org/doc/html/rfc7636>.

**Returns**

[OAuthServiceBuilder](#) the service builder for method chaining

**Supported Clients**

NGClient

**Sample**