

oauth



Apr 04, 2024 11:58

Supported Clients

NGClient

Methods Summary

CustomApiBuilder	customApi(authorizationBaseUrl, accessTokenEndpoint)	Create a custom OAuth api builder.
OAuthService	getOAuthService(provider, clientId, clientSecret, scope, state, deeplinkmethod)	Creates an OAuth service that can be used to obtain an access token and access protected data.
String	getUsedRedirectUrl(builder)	Help method to get the redirect URL which needs to be configured on the OAuth provider application page.
OAuthServiceBuilder	serviceBuilder(clientID)	Creates an OAuth service configurator.

Methods Details

customApi(authorizationBaseUrl, accessTokenEndpoint)

Create a custom OAuth api builder.

Parameters

- String authorizationBaseUrl the base URL where you should redirect your users to authenticate your application
- String accessTokenEndpoint the URL that receives the access token requests

Returns

CustomApiBuilder

Supported Clients

NGClient

Sample

```

var customApi = plugins.oauth.customApi("https://example.com/oauth2/default/v1/authorize", //authorization base
url                                     "https://example.com/oauth2/default/v1/token");    //access token

endpoint
customApi = customApi.
plugins.oauth.serviceBuilder("0lqd1s0aw...") //client/application ID
                                .clientSecret("bIk6163KHi...") //client secret
                                .state("secret123337") //anti forgery session
state, required by the Facebook api
                                .deeplink("deeplink_method") //OPTIONAL deeplink method name or
last part of your redirect URL, see docs
                                                                //if
missing, a global method with the name 'deeplink_svy_oauth' will be generated
                                .callback(myFunction, 30) //see function below, timeout
is 30 seconds
                                .build(customApi); //the custom api created above

function myFunction(result, auth_outcome) {
if (result)
{
    //SUCCESS
    var service = auth_outcome;
    if (service.getAccessToken() == null) return;
    var response = service.executeGetRequest("https://graph.facebook.com/v2.11/me");
    if (response.getStatusCode() == 200) {
        application.output(response.getBody());
        var json = response.getAsJSON();
        application.output("Name is "+json.name);
    }
    else {
        application.output('ERROR http status '+response.getStatusCode());
    }
}
else {
    //ERROR
    application.output("ERROR "+auth_outcome);//could not get access token, request timed out, etc..
}
}
}

```

getOAuthService(provider, clientId, clientSecret, scope, state, deeplinkmethod)

Creates an OAuth service that can be used to obtain an access token and access protected data. This method will be deprecated in the following versions, the preferred way is `plugins.oauth.serviceBuilder` with a callback function.

Parameters

String provider	an OAuth provider id, see <code>plugins.oauth.OAuthProviders</code>
String clientId	your app id
String clientSecret	your client secret
String scope	configures the OAuth scope. This is only necessary in some APIs (like Microsoft's).
String state	configures the anti forgery session state. This is available in some APIs (like Facebook's).
String deeplinkmethod	the name of a global method, which will get the code returned by the OAuth provider

Returns

[OAuthService](#) the OAuthService.

Supported Clients

NGClient

Sample

```

var clientId = "";
var clientSecret = "";
var scope = null;
var state = "secret123337";
var callback = "deeplink";
service = plugins.oauth.getOAuthService(plugins.oauth.OAuthProviders.FACEBOOK, clientId, clientSecret, null,
state, callback, null)
application.showURL(service.getAuthorizationURL());

function deeplink(a,args) {
    service.setAccessToken(args.code);
    var response = service.executeGetRequest("https://graph.facebook.com/v2.11/me");
    if (response.getStatusCode() == 200) {
        application.output(response.getBody());
        var json = response.getAsJSON();
        application.output("Name is "+json.name);
    }
    else {
        application.output('ERROR http status '+response.getStatusCode());
    }
}

```

getUsedRedirectUrl(builder)

Help method to get the redirect URL which needs to be configured on the OAuth provider application page. The url is computed based on what is set on the service builder: deeplink method name, response mode and response type.

Parameters

[OAuthServiceBuilder](#) builder an OAuth service builder

Returns

[String](#) the redirect url

Supported Clients

NGClient

Sample**serviceBuilder(clientID)**

Creates an OAuth service configurator.

Parameters

[String](#) clientID;

Returns

[OAuthServiceBuilder](#) an OAuth service builder object

Supported Clients

NGClient

Sample

```

plugins.oauth.serviceBuilder("0lqd1s0aw...")           //client/application ID
                                .clientSecret("bIk6163KHi...")           //client secret
                                .defaultScope("email")           //ask permission to get
the user email
                                .state("secret123337")           //anti forgery session
state, required by the Facebook api
                                .deeplink("deeplink_method")           //OPTIONAL deeplink method name or
last part of your redirect URL, see docs
                                                                //if
missing, a global method with the name 'deeplink_svy_oauth' will be generated
                                .callback(myFunction, 30)           //see function below, timeout
is 30 seconds
                                .build(plugins.oauth.OAuthProviders.FACEBOOK);

function myFunction(result, auth_outcome) {
if (result)
{
    //SUCCESS
    var service = auth_outcome;
    if (service.getAccessToken() == null) return;
    var response = service.executeGetRequest("https://graph.facebook.com/v2.11/me");
    if (response.getStatusCode() == 200) {
        application.output(response.getBody());
        var json = response.getAsJSON();
        application.output("Name is "+json.name);
    }
    else {
        application.output('ERROR http status '+response.getStatusCode());
    }
}
else {
    //ERROR
    application.output("ERROR "+auth_outcome);//could not get access token, request timed out, etc..
}
}
}

```