

REST API Social login

Implementation instructions

* This plugin makes use of free plugins

[Nextend Social Login and Register](#) and [JWT Authentication for WP REST API](#)

Through an extension of the functionality of the plugins, it creates a **native login system (with user and password)**

and a social login (**facebook and google**) via the **REST API**.

Please note: In order to function properly the two plugins (**NSL and JWT**) must be configured correctly as indicated in the respective documentation.

Here are the instructions to use the native login:

The screens refer to REST requests through the [POSTMAN](#) software

Native login request:

in this case the **endpoint** is that provided by the **JWT** plugin.

You will get a json as a response containing a token, the user email and the username.

The **token** can be used for any protected requests.

In case of protected requests it will have to be set in the Headers.

Authorization: Bearer [token]

as explained in the **JWT plugin documentation**

POST [www.your-domain.com /wp-json/jwt-auth/v1/token?username=alfredo&password=alfredo](http://www.your-domain.com/wp-json/jwt-auth/v1/token?username=alfredo&password=alfredo) Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	alfredo	
<input checked="" type="checkbox"/> password	alfredo	
Key	Value	Description

body Cookies (1) Headers (13) Test Results Status: 200 OK Time: 1938ms Size: 994 B Save Response

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "token":
3     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL2p1c3RCaXRQcm9qXC9iZWRYb2NrXC93ZWliLCJpYXQiOiJlE10DA4OT
4     A3MTUsIm5iZiI6MTU4MDg5MDcxNSwiZXhwIjoxNTgxNDk1NTE1LCJkYXRhIjpb7InVzZXIiOi0",
5     "user_email": "alfred@your-domain.com",
6     "user_nickname": "alfredo",
7     "user_display_name": "alfredo"
8 }

```

Facebook social login:

Facebook access token: [see documentation](#).

BODY payload:

```
{"Access_token": "xxxxxxxx", "token_type", "bearer", "expires_in": 5183946}
```

* in addition to the above parameters, the email must be added, then it becomes:

```
{"Access_token": "xxxxxxxx", "token_type", "bearer", "expires_in": 5183946,
"email", "facebook\_email@domain.com"}
```

POST [www.your-domain.com /wp-json/api/v1/social-login/facebook](http://www.your-domain.com/wp-json/api/v1/social-login/facebook) Send Save

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON Beautify

```

1 {
2   "access_token": "EAAQgM5rZakZAIbAAkviV6ogabjNMWGUcIXaCCdUeZCj1W3aUj0eGI5KCCYZCZC6Dngyx1ZBo8fxhXUxJ4ZA5oVG1UL1IBA8AZAU9NJ2PSiUgplfW4X40FreeeZ
3   aaSSWabZA2lcGmsvdBJcZA2cMg3PDPb0HqptPBGB9r99N09Rc1Mh7eernZBkl0zyv2e2zWP42ggZAQVLZAKukaqgZDZD",
4   "token_type": "bearer",
5   "expires_in": 5183946,
6   "email": "facebook_email@domain.com"
7 }

```

Google social login:

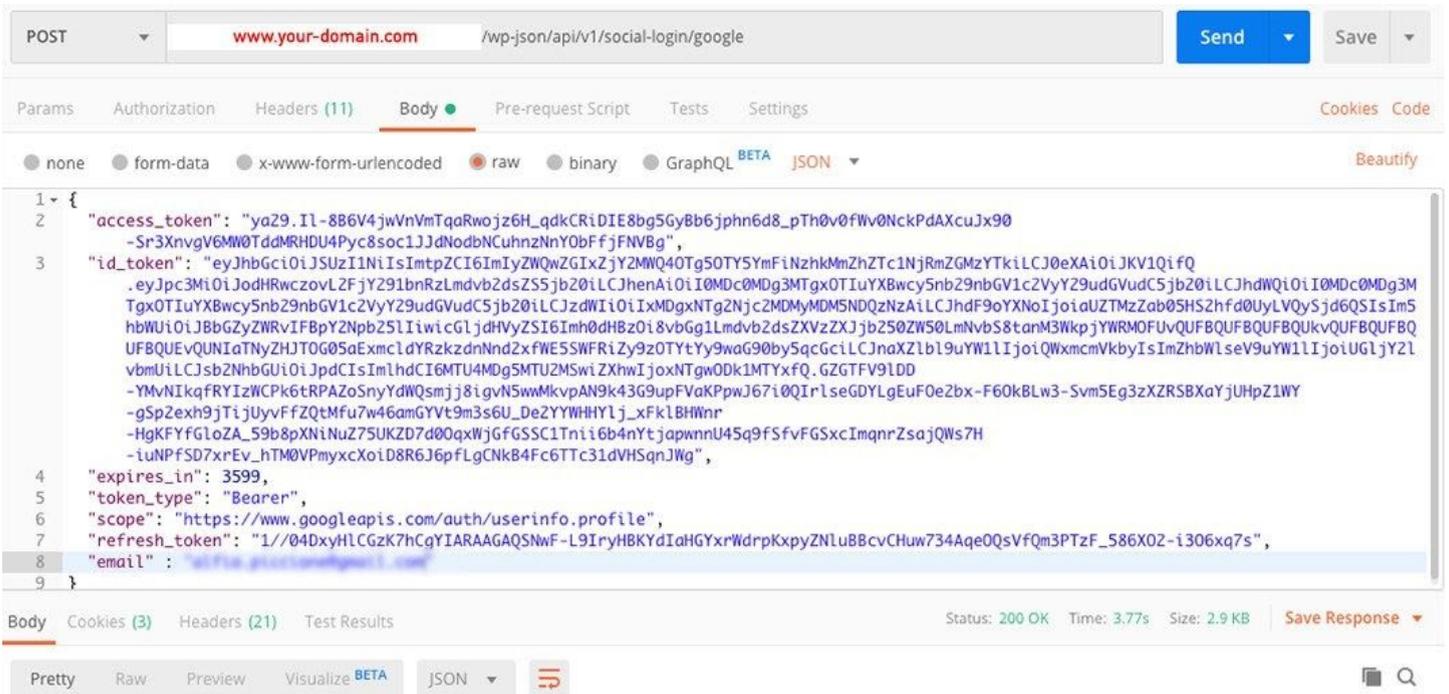
Google token: To get the google [token click here](#).

BODY payload:

```
{"Access_token": "xxxxxxx", "expires_in": 3600, "id_token": "xxxxxxx",  
"token_type": "Bearer"}
```

* in addition to the above parameters, the email must be added, then it becomes:

```
{"Access_token": "xxxxxxx", "expires_in": 3600, "id_token": "xxxxxxx",  
"token_type": "Bearer", "email": "google\_email@gmail.com"}
```



The screenshot shows a REST client interface with a POST request to `www.your-domain.com/wp-json/api/v1/social-login/google`. The body is raw JSON:

```
1 {  
2   "access_token": "ya29.II-8B6V4jwVnVmTqarwojz6H_qdkCRiDIE8bg5Gy8b6jphn6d8_pTh0v0fWv0NckPdAXcuJx90  
   -Sr3XnvgV6MW0TddMRHdu4Pyc8soc1JJdNodbnCuhnzNnYObFfjFNVBg",  
3   "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImIyZWQwZGIxZjY2MmQ0OTg5OTY5YmFiNzhkMmZhc1NjRmZGMzYTkiLCJ0eXAiOiJKV1QiLCJ0eXN0IjoiOiJkbWQwZGZ0eXN0IiwiaWF0IjoiY29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3M  
   .eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJhenAiOiI0MDc0MDg3Mjg0OTIuYXBwcy5nb29nbGV1c2V5Y29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3M  
   Tgx0TiuYXBwcy5nb29nbGV1c2V5Y29udGVudC5jb20iLCJzdWIiOiI0MDc0MDg3Mjg0OTIuYXBwcy5nb29nbGV1c2V5Y29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3M  
   hbWU0iJmVudGVudC5jb20iLCJhdWQiOiI0MDc0MDg3Mjg0OTIuYXBwcy5nb29nbGV1c2V5Y29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3Mjg0OTIuYXBwcy5nb29nbGV1c2V5Y29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3M  
   UFBQUEVUNQAiATNyZjZjY291bnRzLmdvb2dsZS5jb20iLCJhdWQiOiI0MDc0MDg3Mjg0OTIuYXBwcy5nb29nbGV1c2V5Y29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3M  
   vbmUiLCJsb2NhbGU0iJpdCI6ImIyZWQwZGIxZjY2MmQ0OTg5OTY5YmFiNzhkMmZhc1NjRmZGMzYTkiLCJ0eXN0IjoiOiJkbWQwZGZ0eXN0IiwiaWF0IjoiY29udGVudC5jb20iLCJhdWQiOiI0MDc0MDg3M  
   -YmVNIkqfRYIzWCPk6tRPAZoSnyYdWQsmj8igvN5wMkvpAN9k43G9upFVaKppwJ67i0QIrIseGDYLGUeF0e2bx-F60kBLw3-Svm5Eg3zXZRSBXaYjUHPZ1WY  
   -gSp2exh9jTijUyvFfZQtMfu7w46amGYvt9m3sGU_De2YYWHYLj_xFkLbHWnr  
   -HgKfYfGLoZA_59b8pXNiNuZ75UKZD7d00qxWjGfGSSC1Tnii6b4nYtjapwnnU45q9FsvFGSxcImqnrZsajQwS7H  
   -iuNPFSD7xrEv_hTM0VPmyxcXoiD8R6J6pFgCnkB4Fc6TTC31dVHSqnJWg",  
4   "expires_in": 3599,  
5   "token_type": "Bearer",  
6   "scope": "https://www.googleapis.com/auth/userinfo.profile",  
7   "refresh_token": "1//04DxyHLCgZ7hCgYIARAAGASNWf-L9IryHBKYdIahGYxRwdrpKxpyZNLuBBcvCHuw734Aqe0QsVfQm3PtZf_586X02-i306xq7s",  
8   "email": "google\_email@gmail.com",  
9 }
```

The status bar shows: Status: 200 OK, Time: 3.77s, Size: 2.9 KB. The response is displayed in JSON format.

Login request response:

The response of the social login request will be identical to that of the native login. Also in this case the **token** can be used for any protected requests.

```
Body Cookies (3) Headers (21) Test Results Status: 200 OK Time: 3.77s Size: 2.9 KB Save Response
Pretty Raw Preview Visualize BETA JSON
1 {
2   "token":
3     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOmwvXC9sb2NhbGhvc3RcL2p1c3RCaXRQcm9qXC9iZWRYb2NrXC93ZWlilCJpYXQ0jE10DA40T
4     E2MDEsIm5iZiI6MTU4MDg5MTYwMSwiZXhwIjoxNTgxNDk2NDAxLCJkYXRhIjpb7InVzZXIiOi0nsiaW0iOjkwfX19.
5     5hEVHiV-PoWT6zruytdZk2_eS_kSVkBwdShufhz42_U",
6   "user_email": "alfred.piccione@wpdev.it",
7   "user_nickname": "google-alfredopiccione",
8   "user_display_name": "Alfredo Piccione"
9 }
```

Any checks for requests not protected by **JWT** can be done through the filter **\$noAuth = apply_filters('sl_extended_no_auth_request', \$noAuth, \$request, \$_SERVER['REQUEST_URI'], \$_SERVER['REQUEST_METHOD']);** in the **checkNoAuthRequest()** method of the RestApi class.

Decode token:

This endpoint allows you to decode the token and retrieve the user id

BODY payload:

```
{"token": "xxxxxxx"}
```

example:

```
POST http://localhost/wpdev/wp-json/api/v1/decode Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOmwvXC9sb2NhbGhvc3RcL3dwZGV2IiwiaWF0IjoxNTgxMjE0IiwiaW0iOjkwfX19.
3   o2u5g2hzo9G-wDgBAKdFRX10tLePARiHo9pyBDNkseyY"
4 }
Body Cookies (1) Headers (15) Test Results Status: 200 OK Time: 2.61 s Size: 826 B Save
Pretty Raw Preview Visualize JSON
1 {
2   "iss": "http://localhost/wpdev",
3   "iat": 1592299118,
4   "nbf": 1592299118,
5   "exp": 1592903918,
6   "data": {
7     "user": {
8       "id": 6
9     }
10  }
11 }
```