

OPEN API OAuth 2.0 Authentication and Authorization API Specification

Document version: v1.1

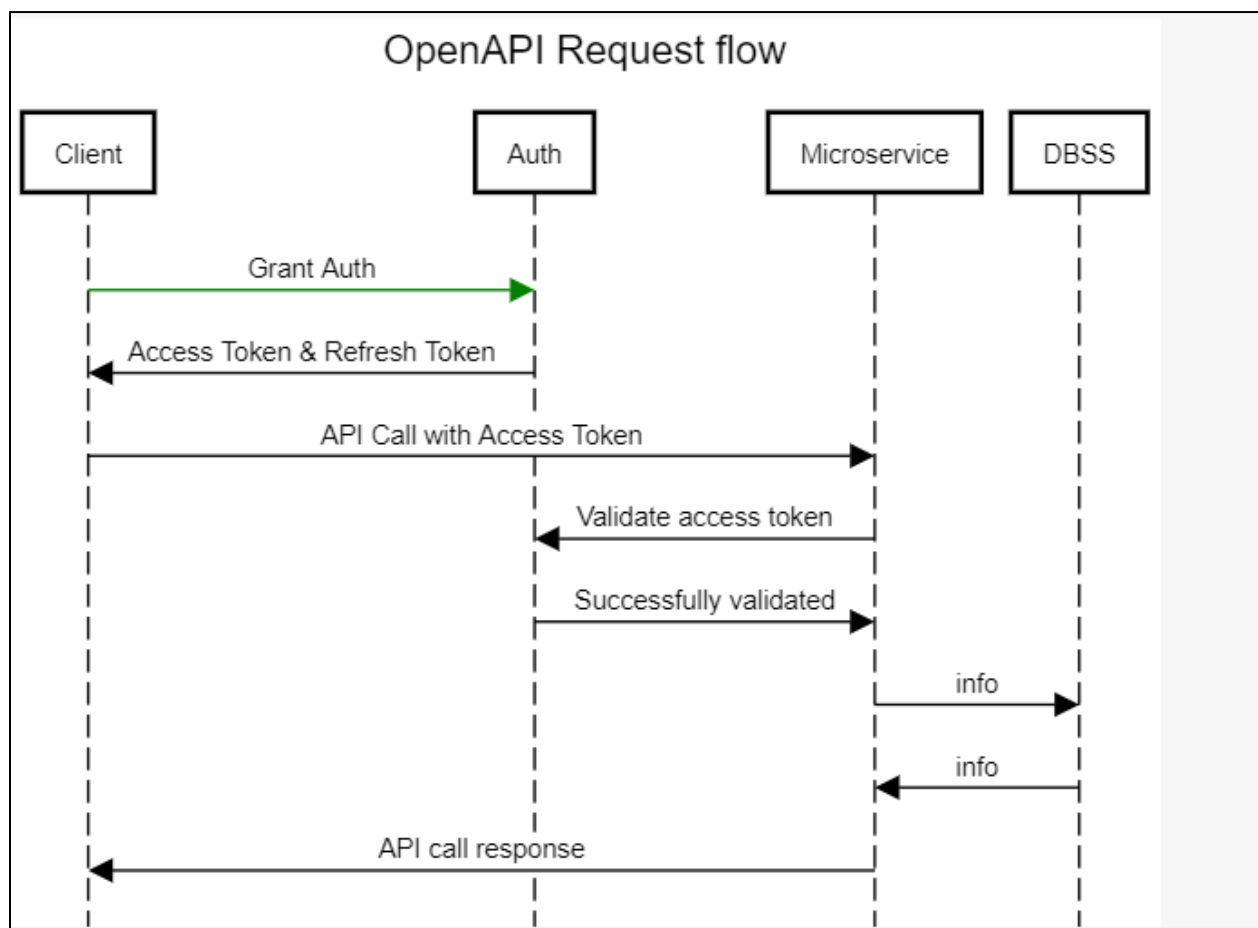
Contents

About the Document	3
Process Flow	3
Access token	4
End point	4
Authorization	4
Request header	4
Request body	5
Request:	6
Response	6
Refresh Token	7
End point	7
Authorization	8
Authorization	8
Request header	8
Request body	8
Response	9
References:	10
OAuth 2.0:	10
Link to generate token:	10
Decode JWT token data:	11

About the Document

This document is intended to provide specification for Authentication and Authorization API of Banglalink.

Process Flow





Access token

End point

<http://<IP>:<Port>/auth/realms/banglalink/protocol/openid-connect/token>

Test URL & Production URL: Will be provided by Banglalink

Request Type: POST

Authorization

Type: Basic Auth

Username: <client_id>

Password: <client_secret>

Note: Client ID and secret will be shared by Banglalink .

POST ⌵ `{{host}}/auth/realms/banglalink/protocol/openid-connect/token` Send ⌵

Params **Authorization** ● Headers (11) ● Body ● Pre-request Script Tests Settings Cookies

Type Basic Auth ⌵

ⓘ Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#) ✕

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#) ↗

Username

Password

☐ Show Password

Figure: Screenshot from postman.

Request header

key	value
Content-Type	application/x-www-form-urlencoded
Authorization	Basic <token>



For checking the API with postman `<client_id>` and `<client_secret>` can be put in Authorization tab to generate basic auth token. Basic auth token can be generated from below URL to put in header section while calling the API.

Basic Auth: base64(Client ID:Client Secret)

Link to generate token: <https://mixedanalytics.com/tools/basic-authentication-generator/>

Request body

key	value	Mandatory	Remarks
grant_type	password	yes	Fixed value
username	<code><username></code>	yes	Will be shared by Banglalink
password	<code><password></code>	yes	Will be shared by Banglalink
scope	openid	no	
client_id	<code><client_id></code>	yes	Will be shared by Banglalink

Figure: Sample request with postman

POST `{{host}}/auth/realms/banglalink/protocol/openid-connect/token` Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
☒ grant_type	password			
☒ username	XXX			
☒ password	XXX			
☒ scope	openid			
☒ client_id	XXX			
Key	Value	Description		

Request:

HTTP request sample

```
POST /auth/realms/banglalink/protocol/openid-connect/token HTTP/1.1
Host: <ip>:<port>
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <token>
grant_type=password&username=<username>&
password=<password>&scope=openid&client_id=<client_id>
```

CURL request sample

```
curl --location --request POST '<ip>:<port>/auth/realms/banglalink/protocol/openid-connect/token' \
--header 'Authorization: Basic <token>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=<username>' \
--data-urlencode 'password=<password>' \
--data-urlencode 'scope=openid' \
--data-urlencode 'client_id=<client_id>'
```

Response

key	type	Remarks
access_token	string	authentication token for API call

expires_in	int	36000 seconds
refresh_token	string	Will be used to generate refresh token
refresh_expires_in	int	36000 seconds

Other parameters will not be required.

```
Http 200 Response
{
  "access_token": "XXX",
  "expires_in": 36000,
  "refresh_expires_in": 36000,
  "refresh_token": "XXX",
  "token_type": "Bearer",
  "id_token": "XXX",
  "not-before-policy": 0,
  "session_state": "XXX",
  "scope": "openid email profile"
}

Http 401 Response
{
  "error": "invalid_grant",
  "error_description": "Invalid user credentials"
}

Http 400 Response
{
  "error": "unsupported_grant_type",
  "error_description": "Unsupported grant_type"
}
```

Refresh Token

End point

<http://<IP>:<Port>/auth/realms/banglalink/protocol/openid-connect/token>

Test URL & Production URL: Will be provided by Banglalink

Request Type: POST



Authorization

Authorization

Type: Basic Auth

Username: <client_id>

Password: <client_secret>

Note: Client ID and secret will be shared by Banglalink .

Request header

key	value
Content-Type	application/x-www-form-urlencoded
Authorization	Basic <token>

For checking the API with postman <client_id> and <client_secret> can be put in Authorization tab to generate basic auth token. Basic auth token can be generated from below URL to put in header section while calling the API.

Basic Auth: base64(Client ID:Client Secret)

Link to generate token: <https://mixedanalytics.com/tools/basic-authentication-generator/>

Request body

key	value	Mandatory	Remarks
grant_type	refresh_token	yes	Fixed value
refresh_token	Refresh token received from access token API	yes	
client_id	<client_id>	yes	Will be shared by Banglalink

POST [{{host}}/auth/realms/banglalink/protocol/openid-connect/token](#) Send

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

☒	grant_type	refresh_token	
☒	refresh_token	XXX	
☒	client_id	XXX	
	Key	Value	Description

Response

HTTP request sample:

```
POST /auth/realms/banglalink/protocol/openid-connect/token HTTP/1.1
Host: <ip>:<port>
Content-Type: application/x-www-form-urlencoded
Authorization: Basic XXX
grant_type=refresh_token&refresh_token=<token>&client_id=<client_id>
```

Response

key	type	Remarks
access_token	string	authentication token for API call
expires_in	int	36000 seconds
refresh_token	string	Will be used to generate refresh token
refresh_expires_in	int	36000 seconds

Other parameters will not be required.

```
Http 200 Response
{
  "access_token": "XXX",
  "expires_in": 36000,
  "refresh_expires_in": 36000,
  "refresh_token": "XXX",
  "token_type": "Bearer",
  "id_token": "XXX",
  "not-before-policy": 0,
  "session_state": "XXX",
  "scope": "openid email profile"
}

Http 401 Response
{
  "error": "invalid_grant",
  "error_description": "Invalid user credentials"
}

Http 400 Response
{
  "error": "unsupported_grant_type",
  "error_description": "Unsupported grant_type"
}
```

References:

OAuth 2.0:

<https://www.oauth.com/oauth2-servers/access-tokens/>

<https://www.oauth.com/oauth2-servers/access-tokens/password-grant/>

<https://www.oauth.com/oauth2-servers/access-tokens/refreshing-access-tokens/>

Link to generate token:



<https://mixedanalytics.com/tools/basic-authentication-generator/>

Decode JWT token data:

<https://jwt.io/>