FINANCIAL TRANSMISSION RIGHTS

FTR: API Integration Guide

1 December 2021

Vers. 0.2

Table of Contents

Document History

Version	Date	Status	Edited By	Revision Description
0.1	01 July 2020	Initial Draft	EMS	Initial Draft of Register API
0.2	1 December 2020	Published	E Oosterbaan	API documentation

FTR API Integration Guide

Background

The APIs available as part of the FTR Manager service are used for participants in their business to business APIs.

FTR URLs

The base API URL will be referred to as **{URL}** for the remainder of this guide.

UAT Endpoint:	**{ftr_url} =** https://uat-api.ftr.co.nz
PROD Endpoint:	**{ftr_url}** = https://api.ftr.co.nz

Authorisation

Authentication overview

Authentication for the FTR APIs is handled by User Pools configured within AWS Cognito. Each user within Cognito will have groups assigned to them which help to control what data is returned and which APIs they are allowed to access. The API is served by AWS API Gateway, which checks the user provided token is valid, authenticated and was issued by the correct User Pool. Once API Gateway has validated the token with Cognito to make sure it is correct, it then passes the token back to the API Backend.

The Authorisation and filtering logic for the data to be returned is controlled by the backend. The Cognito tokens that are passed back are in a JWT encrypted format, so they include information about what groups the user belongs to and other user specific attributes like email. Once the backend has filtered the returned data based on the information in the token, the request is passed back through the API Gateway and back to the user.

Authentication process

Step one: Authenticate to AWS Cognito

The first step in the authentication flow is for the end user to authenticate to AWS Cognito with a valid username and password to retrieve a token to pass to API Gateway. In order to authenticate to Cognito, you will need:

- A valid Application Client ID from the userpool
- A valid username and password from the userpool

There are two different broad methods of authentication to Cognito. You can either use one of the AWS SDKs or you can use authenticate based on standard OAuth2 authentication. We recommend wherever possible that the AWS SDK should be used, as this makes the process a lot easier. Cognito tokens, by default, are valid for 60 minutes. There are a number of JWT libraries that can be used to help manage expiry and refresh of tokens as required.

Authenticating via AWS SDK

In order to authenticate with the AWS SDK, you will need to integrate that with your code base. AWS SDKs are developed for the majority of programming languages, and most languages will have sample apps available (i.e. Java and .NET). You will need to call Initiate_auth (Name may change depending on SDK language) method which should take a username and password as input and output a JSON payload containing a series of tokens (IdToken, AccessToken and RefreshToken). The method should not require admin, or IAM, credentials (There's normally options for both admin and normal - in this case, we want normal). Authentication flow should be 'USER_PASSWORD_AUTH'.

For example (but not limited to);

```
import com.amazonaws.services.cognitoidp.AWSCognitoIdentityProvider;
import
com.amazonaws.services.cognitoidp.AWSCognitoIdentityProviderClientBuilder;
import com.amazonaws.services.cognitoidp.model.*;
import java.util.HashMap;
import java.util.Map;
public class DemoApplication {
public AWSCognitoIdentityProvider getAmazonCognitoIdentityClient() {
return AWSCognitoIdentityProviderClientBuilder.standard()
.withRegion( "ap-southeast-2" )
.build();
}
public void authenticate() {
final Map<String, String> authParams = new HashMap<>();
authParams.put( "USERNAME" , "UsernameHere" );
authParams.put( "PASSWORD" , "PasswordGoesHere" );
final InitiateAuthRequest authRequest = new InitiateAuthRequest();
authRequest.withAuthFlow(AuthFlowType.USER_PASSWORD_AUTH)
.withClientId( "2psmqhb*****************6b2q2" )
.withAuthParameters(authParams);
InitiateAuthResult initiateAuthResult =
getAmazonCognitoIdentityClient().initiateAuth(authRequest);
System.out.println( "Authenticate result: " + initiateAuthResult);
}
public static void main(String[] args) {
DemoApplication demoApplication = new DemoApplication();
demoApplication.authenticate();
}
}
```

Authentication for B2B without SDK

You can authenticate with the Cognito service without using the SDK using the following technique.

Request format and instructions

Endpoint:	https://api.ftr.co.nz/auth or in UAT https://uat-api.ftr.co.nz/auth
Method:	POST

Example Payload

```
{
    "AuthParameters": {
        "USERNAME": "user_email_address_here",
        "PASSWORD": "password_here"
    },
    "AuthFlow": "USER_PASSWORD_AUTH",
    "ClientId": "2lgih8……………………………ubu5p"
}
```

usernamePlaceholder = email address used to authenticate to the webapp
passwordPlaceholder = password created on account setup
clientIdPlaceholder provided by EMS

Successful response format

HTTP Code: 200

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJraWQi…",
        "ExpiresIn": 3600,
        "IdToken": "eyJraWQi…",
        "RefreshToken": "eyJjdHk…",
        "TokenType": "Bearer"
    },
    "ChallengeParameters": {}
}
```

All 3 tokens (Access, ID and Refresh) are in JWT encoded format. The '**IdToken**' is the one you need to supply to API Gateway as below. 'ExpiresIn' is the time, in seconds, that the token is valid for.

Error response format

HTTP Code: 400

```
{
"__type": "Exception type"
"message": "Exception message"
}
Example:
{
"__type": "NotAuthorizedException"
"message": "Incorrect username or password"
}
```

Curl example

```
curl -X POST --data @auth-data.json https://api.em6.co.nz/auth
```

Where auth-data.json is the following;

```
{
"AuthParameters" :
{ "USERNAME" : "yourusername@example.com", "PASSWORD" : "yourpassword" },
"AuthFlow" : "USER_PASSWORD_AUTH",
"ClientId" : "AFDG........................"
}
```

Step two: Authenticate to API Gateway

Using the Identity Token (IdToken), we can now authenticate to API Gateway. Simply append the body of the identity IdToken to 'Authorization' header to all your requests. API Gateway will validate with Cognito that the token is valid, not expired and that the token was issued by the correct User Pool. Once this is done, the request will be passed to the backend and data returned (based on what the authenticated Cognito user has access.

FTR Register API

Description: API to return FTR information from the FTR Register.

URL:	**{URL}/ftr_register/[marketName][currentOwner][ftrPeriod][status][settled][recent][page]**
Method:	GET
Query parameters:	
marketName	**PRI_JUN_2021** – *Optional blank for whole registry **NOTE**: see pagination parameter below*
currentOwner	**CTCT [csv]–** four letter FTR participant code. *Optional blank for all participant information.*
ftrPeriod	**01/09/2021 –** start date of the FTR period. *Optional blank for all FTR periods*
status	**AW or AS –** Awarded or Assigned. *Optional blank for all FTR statuses*
settled	**TRUE or FALSE –** If settled=TRUE return all FTRs where start date is < previous month, if settled=FALSE return all FTRs where start date is >= current month *Optional*
recent	**Y –** If recent=Y return all FTRs from most recent dateAcquired field in DB *Optional*
page	**1 –** The FTR register is a large payload, with a maximum of 10K rows per response. The responses are paginated. The response has a total at the end of the response to let the user know if another page is required. **Eg**: {ftr_url}/ftr_register/?page=1 will return the first 10K rows of the FTR register.

Examples:

1. {ftr_url}/ftr_register/
2. {ftr_url}/ftr_register/?marketName=PRI_JUN_2021
3. {ftr_url}/ftr_register/?marketName=PRI_JUN_2021&currentOwner=CTCT,MERI
4. {ftr_url}/ftr_register/?settled=FALSE
5. {ftr_url}/ftr_register/?recent=Y
6. {ftr_url}/ftr_register/?page=1
7. {ftr_url}/ftr_register/?page=2

Example Truncated FTR Register API Response:
{ftr_url}/ftr_register?marketName=VAR_NOV_2021

```
[
    {
        "marketName": "VAR_NOV_2021",
        "dateAcquired": "18/11/2021",
        "status": "AWARDED",
        "ap": "2018",
        "awards": [
            {
                "startDate": "01/12/2021",
                "endDate": "31/12/2021",
                "source": "INV",
                "sink": "WKM",
                "hedgeType": "OPT",
                "ftrs": [
                    {
                        "ftrId": 5565360,
                        "currentOwner": "CNIR",
                        "previousOwner": null,
                        "firstOwner": "CNIR",
                        "mw": 1.0,
                        "price": 11.85,
                        "aq": 8816.4,
                        "origAq": 8816.4
                    }
                ]
            },
            {
                "startDate": "01/07/2022",
                "endDate": "31/07/2022",
                "source": "ISL",
                "sink": "BEN",
                "hedgeType": "OPT",
                "ftrs": [
                    {
                        "ftrId": 5563447,
                        "currentOwner": "CNIR",
                        "previousOwner": null,
                        "firstOwner": "CNIR",
                        "mw": 50.0,
                        "price": 0.0,
                        "aq": 0.0,
                        "origAq": 0.0
                    },
```

Example Truncated FTR Register Assignments API Response: {ftr_url}/ftr_register?status=as

```
[
    {
        "marketName": null,
        "dateAcquired": "22/10/2013",
        "status": "ASSIGNED",
        "ap": "2012",
        "awards": [
            {
                "startDate": "01/12/2013",
                "endDate": "31/12/2013",
                "source": "BEN",
                "sink": "OTA",
                "hedgeType": "OPT",
                "ftrs": [
                    {
                        "ftrId": 200761,
                        "currentOwner": "OMFM",
                        "previousOwner": "MERI",
                        "firstOwner": "MERI",
                        "mw": 12.0,
                        "price": 6.33,
                        "aq": 56514.24,
                        "origAq": null
                    }
                ]
            }
        ]
    },
```