|  |
| --- |
| RFGLogoFinal-Color.jpg |
| RFG REST API(s) – OAuth 2.0 Client Credentials Grant Guide |
| How to access Okta protected RFG REST API(s) using OAuth 2.0 Client Credentials Grant |
| **Version: 1.0** |
|  |

Business Technology Team

*Supporting the Business*

**Version History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version # | Date | Section # | Action | Description | Author(s) | Document Reviewed By |
| 1.0 | 06/14/2018 | All | Add | Initial Creation of document | Sami Abdul | Mark Milchuk, Gerry Mayer |
| 1.0 | 07/02/2019 | All | Update | Apigee version | Irem Fiza | Shshank Mathur,  Vinod Athiyarath |

Table of Contents

[1 Introduction 1](#_Toc516751546)

[1.1 Terms and Abbreviations 2](#_Toc516751547)

[1.2 References 3](#_Toc516751548)

[2 Client Registration 4](#_Toc516751549)

[3 RFG Authorization Server Information 5](#_Toc516751550)

[4 Additional Instructions for Developers 5](#_Toc516751551)

[5 Postman Client 6](#_Toc516751552)

[6 SOAP UI Client 10](#_Toc516751553)

# Introduction

This document describes how to use OAuth 2.0 client credentials grant to acquire access token from RFG Authorization Server, and how to use it with the protected RFG REST API(s). The Client Credentials grant is used when application request an access token to access their own resources, not on behalf of user. The access token will be for the “application”, an application context not a user context.

High Level Steps (see Figure 1):

1. A client application sends its own credentials (Client ID, Client Secret and Scope) to RFG Okta Authorization Server 🡪 Token Endpoint (Access Token URL).
2. RFG Authorization server validates the application credentials.
3. If the credentials are valid, RFG Authorization Servers returns an access token to the client application with a scope set to the “App ID URI” of the requested RFG REST API (Web API).
4. Now, with a valid access token (access\_token) and apiKey for the Apigee API Management Portal ,the client application will call protected RFG API.
5. Apigee API Management gateway validates the apiKey and the access token.
6. RFG API returns the result (JSON object).

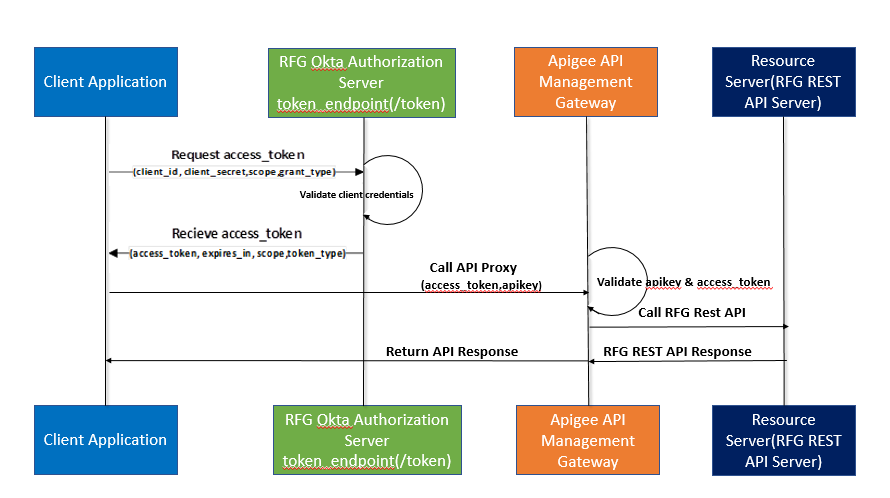


Figure : High level steps for calling RFG REST API(s) using OAuth 2.0 Client Credentials Grant

Note: In this document, several screenshots are included to guide you how to integrate your client application with RFG API(s). Some fields are blurred in the screenshots as they are Vendor specific (or environment specific). RFG will provide values for those fields separately (for Non-Prod and Production environments).

## Terms and Abbreviations

| Term/Abbreviation | Definition |
| --- | --- |
| BTT | Business Technology Team |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |
| RFG | Realogy Franchise Group |
| REST | Representational State Transfer |
| Client Credentials Flow | The Client Credentials flow is recommended for use in machine-to-machine authentication. Your application will need to securely store it’s Client ID and Secret and pass those to Realogy Okta in exchange for an access token. |
| Access Token (access\_token) | Access tokens are credentials used to access protected resources. An access token is a string representing an authorization issued to the client. Access tokens are used as bearer tokens. A bearer token means that the bearer can access authorized resources without further identification.  Note: For RFG Authorization Server, it will be a JSON Web Token (JWT) |
| Claim | Piece of information asserted about an Entity. Simply put, claims are Name/value pairs that contain information about a user. |
| Issuer | Entity that issues a set of Claims. |
| Scope | Scopes are space-separated lists of identifiers used to specify what access privileges are being requested |
| "iss" (Issuer) Claim | The issuer of the token (JWT), an entity we trust. |
| "sub" (Subject) Claim | The subject claim identifies the principal that is the subject of the JWT. |
| "aud" (audience) Claim | The audience claim identifies the recipients that the JWT is intended for. |
| "exp" (expiration time) claim | The expiration time claim identifies the expiration time on or after which the JWT MUST NOT be accepted for processing. |

## References

| Title | Link |
| --- | --- |
| The OAuth 2.0 Authorization Framework | <https://tools.ietf.org/html/rfc6749> |
| JSON Web Token (JWT) | <https://tools.ietf.org/html/rfc7519> |
| Implementing the Client Credentials Flow | <https://developer.okta.com/authentication-guide/implementing-authentication/client-creds> |
| Decode JSON Web Tokens | <https://jwt.io/> |
| Postman – API Testing Tool | <https://www.getpostman.com/> |
| SoapUI – API Testing Tool | <https://www.soapui.org/> |
|  |  |

# Client Registration

First, you need to register the client application with RFG. After registeration process is completed, you will be provided client credentials, api key, scope(s) and RFG REST API URL(s) for your client application. Each client application must have separate credentials set and api key.

After client registration, you should receive below values from RFG for calling RFG API(s).

|  |  |
| --- | --- |
| Client ID | It will be provided for your client application. It is the unique identifier of your application. It will be used to get access token. |
| Client Secret | It will be provided for your client application. It is the key that you pass with Client ID to token endpoint to get access token. |
| List of App ID URIs for RFG REST APIs (for scope parameter) | It will be provided for your client application based on your application asccess to RFG REST APIs.  For Example: App ID URI for Dash API  <https://btt.realogyfg.com/dashapi>  Scope: A space-separated list of App ID URI(s) for Realogy API(s). Scopes enable your application to access specific RFG REST APIs.  Note: App ID URI is **NOT** where the REST API is hosted, it is instead a unique identifier of the REST API. |
| RFG REST API URL(s) | RFG REST API URL(s) will be communicated separately for each environment – Non-Prod and Production.  Note: This is where the RFG REST API(s) are hosted. |
| Api Key | Every call to the RFG API(s) requires an api key. This key needs to be passed in the request header (**apiKey**). |

# RFG Authorization Server Information

**Non-Prod RFG Okta Metadata Document:**

<https://realogy.oktapreview.com/oauth2/ausdtpyw647fbrcPi0h7/.well-known/oauth-authorization-server>

**Non-Prod RFG Access Token URL (token\_endpoint):**

<https://realogy.oktapreview.com/oauth2/ausdtpyw647fbrcPi0h7/v1/token>

**Grant Type:** client\_credentials

**Scope:** App ID URI of REST API e.g. <https://btt.realogyfg.com/dashapi>

# Additional Instructions for Developers

|  |  |
| --- | --- |
|  | Instruction |
| 1. | All access tokens have an expiration and must be cached by the application within 5 minutes of expiration. Applications who fail to cache their access tokens and request new access tokens for every REST API call they make will have their access immediately suspended. |
| 2. | Always store the client secret key securely. If you suspect that the secret key has been compromised, then you can request for another client secret. This should never be stored in a Mobile App or Native App. |

# Postman Client

Here are step-by-steps instructions to consume a protected RFG API using Postman Client.

|  |  |
| --- | --- |
|  | Step |
| 1. | **Request Access Token**  Set Authorization Type to OAuth 2.0 and get new access token using client\_credentials grant type.  Access Token URL: Provide token\_endpoint  e.g Non-Prod RFG Access Token URL:  <https://realogy.oktapreview.com/oauth2/ausdtpyw647fbrcPi0h7/v1/token>  Client ID: Enter client\_id provided to your client application  Client Secret: Enter client\_secret provided to your client application  Scope: Enter specific App Id URI assigned to RFG API that you want to call  e.g. <https://btt.realogyfg.com/dashapi> (Dash API)  Client Authentication: Send client credentials in body    **Receive Access Token (Succesful Response):**  If request for access token is valid (client credentials and scope values are good) then Authorization server will generate access\_token along with additional properties (e.g. token\_type, expires\_in and scope)  Here is an explanation of values returned in a successful response:   * access\_token : The access token string as issued by the authorization server. * token\_type : The type of token this is, typically just the string “bearer” * expires\_in (seconds): Duration of time the access token is granted for. * scope: App ID URI assigned to RFG API. It should be identical to the scope the client application requested. |
| 2. | **Decode Access Token (optional):**  If you decode the access\_token (use <https://jwt.io> or similar online tools to decode the token), you will see it contains following claims: |
| 3. | **Set Headers:**  Set access\_token (received in step-1) and apikey (assigned to your client application) as headers (key, value pairs)  apiKey: {Api Key assigned to your application}  Authorization: Bearer {access\_token received in previous step}  Note: You can manually set access\_token in Authorization header or you can select Authorization Type as OAuth 2.0 (and select access\_token acquired in previous step) |
| 4. | **Call protected RFG API:**  Call RFG API with required parameters.  For Example: Dash API  [https://api.realogy.com/1.0/dash/batches/{batchId}](https://api.realogy.com/1.0/dash/batches/%7bbatchId%7d)  It will return JSON object as response. |

# SOAP UI Client

Here are step-by-steps instructions for SOAP UI client.

|  |  |
| --- | --- |
|  | Step |
| 1. | **Create New REST Project:**  Open SOAP UI and create a new REST Project and give the url as https://{apiurl} (this RFG API url will be communicated separately for each RFG API).  For Example: Dash API (QA)  <https://api.realogy.com/v1.0/dash/batches/11055> |
| 2. | **Add New Authorization:**  Type: OAuth 2.0  Profile Name: Enter any name for this profile e.g. RFG Client Credentials  Click on “Get Token” |
| 3. | **Get Access Token from Authorization Server:**  Select OAuth 2 Flow: Client Credentials Grant  Client Identification : Enter client\_id provided to your client application  Client Secret : Enter client\_secret provided to your client application  Access Token URI: Provide token\_endpoint  e.g Non-Prod RFG Access Token URL:  <https://realogy.oktapreview.com/oauth2/ausdtpyw647fbrcPi0h7/v1/token>  Scope: Enter specific App Id URI assigned to RFG API that you want to call  e.g. <https://btt.realogyfg.com/dashapi> (Dash API)    **Sueccessful Response:**  If request for access token is valid (client credentials and scope values are good) then Authorization server will generate access\_token |
| 4. | **Decode Access Token (Optional):**  If you decode the access\_token (use <https://jwt.io> or similar online tools to decode the token), you will see it contains following claims: |
| 5. | **Set Headers:**  Add apiKey to headers.  apiKey: {Api Key assigned to your application}  Note: SOAP UI client will automatically send access\_token as Authorization header |
| 6. | **Call RFG API:**  Submit request to protected Realogy API (with access token and api key as headers) |