# Redkik 3 API usage

Oct, 2023: Updated for Redkik 3.0

# Introduction

Redkik has launched its third iteration of the Redkik platform, appropriately known as Redkik 3.0. This guide quickly walks through the basic concepts and basic actions available through the Redkik RESTful API and common patterns.

#### Concepts

#### Bookings

A Booking represents coverage of a specific shipment. The steps of creating a quote and then purchasing it are simple:

- 1. Generate a quote by POSTing a json object of the following data:
  - a. Fields
    - Commodity information
    - Journey information
    - Cargo Owner information
  - b. retain the ID of the quote, returned with a successful quote
- 2. then use that ID to purchase the offer, which creates a booking.

The Lifecycle of a Booking is:

- 1. Setup
- 2. Quote
- 3. Purchase
- 4. Cancel, or
- 5. Complete

# **API** Tokens

Redkik leverages OAUTH 2.0 for authentication strategies. To learn more about oAuth2, please go to:

#### OAuth 2.0 — OAuth

The authentication flow is as follows:

- 1. Generate API Tokens for an organization, typically Redkik's job.
- 2. Generate a onetime, limited-lifespan access token for endpoint access
- 3. Refresh Access Tokens for continued use of that authentication session

#### Generate API Tokens

API Tokens are generated for an organization, consist of a client\_id and a secret\_id keypair, and are used to generate one-time access tokens with a limited time-to-live.

Browse to <u>Organizations</u>, select your organization (or a child-org if creating a Quote "On Behalf Of" that child-org), right click on the org's config button, and select API Tokens:



Add a new API Token:

💼 🖹 Redik   App 🛛 🖈 🗈 Redik   App	× +						-	ø ×
← C	ons/1565148d-d618-4254-a271-a8dcccd4fe7b/a	piTokens/create				A* 16 12	@ Ø	<del>8</del> "•
	Redkik			💶 💼  🎆 Tait Truc	king <del>-</del>			Î Q
	•			- 10				
Be	lookings Booking Holders Organiza	tions						٩
•	Home / Organizations / Redkik Truckin	g Gap - API Tokens / Create						5
	*easta							+
	Neare							
N	Name							
	<ul> <li>Test Token</li> </ul>				5			
0	organization.apiTokens.field.role							
	? Customer Admin	✓ Quote	<ul> <li>Booking Admin</li> </ul>	Organization Admin	5			
				X Cancel A Su	bmit			
							_	_
							?	œ
							-	. 🕸

Upon Submission, you will have access to the client\_id and the secret\_id. Please ensure you save these off as this will be the only time you have access to them!



Test it out in Swagger, using the <u>/oauth/token</u> endpoint:

/api/v2/user/oauth/token	^ 🗈
arameters	Cancel Reset
ame Description	
uthorization authorization header)	
equest body	application/json
<sup>1</sup> <sup>4</sup> <u>szart 1728'' "článt Levészila</u> " <u>* článt 14'' véss240-052-0562-0562-070446936582*.</u> <u>* článt 1- véss240-0552-0562-0562-05666666666666666666666</u>	

Producing an Access token and Refresh token:

C (A)	x     Image: The Market Appo     x     Image: The Market Appo     x     +       https://safec.app.red/sk.com//spl/v2/printsprimury/Name-Lise/Service/Puthentication/cauthAccessToken/Request     A     A     A	6) ¢ @	- 1 © 6	· · ·
Curl -X ' 'https: -H 'acc -H 'Com -d 'C "grant_" "client }'	P057 \ /N238x.gpp.red%iK.com/gpi/v2/user/oauth/token' \ /P21.application/json' \ tent-Type: application/json' \ type': "slast_credentals", "" "slast_credentals", "" "slast_credentals.e%2.edd4d98682794551b334c2372275e8a1538" 	ß		Q ♦ ¶ ₩ +
Request UR	2			
https://s	sales.app.redkik.com/api/v2/user/oauth/token			18
Server resp	onse			
Code	Details			
200	Response body {     faither_type*: "Basen",         *secse_type*: "Same",         *secse_type*: "Same",         *secse_taken",         *secses_taken",         *secsesses_taken",	izjyilcJjaNqioi UWMYilcJjaNqio Download		l
	Response headers cochex-control: no-store context-type: application/json; charset=utf-8 dete: Tup, 20 bc:2022 2108.03 G GMT etag: W/*27b-PHE23D#PEC5UGHwGGWUIk+Ocg" pragma: no-coche x-restlaist-remaining: 5 x-restlaist-remaining: 5 x-restlaist-remaining: 5 x-restlaist-remaining: 5			
Responses				
Code	Description	Links		e
200		Mo linke		•
C C	x         D         Pentile   App         x         0. Swegare UI         x         +           https://sales.app.rediki.com//col/v2/poloc/Turk.prmaryName-UserSinice#Authentications/cauthAccessTokenReguest         A <sup>1</sup> 0. r/l		- 1	0 X
		18 1≣ V⊞	69 E	2
Curl -X '' 'https: -H 'com -d '{ "grant_' "client }'	POST \ //sles.spp.redhik.com/api/v2/user/oauth/token' \ app: appl.antion/jsen' \ type': "client_credentials", _L4": "displated-dB3c-dB9c.seds2-efd6dd938682", _ueret': "absplated-JB3c1859c1850-c1850-de86d68c37948b1b334c2372275eb1538"	© ™ @		- - - - - - - - - - - - - - - - - - -
Curl curl -X '' 'https:, -H 'acc: -H 'Com -d '{ "grant_" "client, 'client }' Request UR	P091 \ //sles.spp.redhik.com/api/v2/user/oauth/taken' \ app: appLantim/jsen' \ type': 'Client_credential'', _L4'' "\$5524:0-d93:-499c.4662-4fd4d938882'', _L4'' "\$165524:0-d93:-499c.4662-4fd4d938882'', _ercet': "al6724f209772596cli65x4f2bc6866666:37948blb334c2372278e8a1538'' 3L	·• · · · · · · · · · · · · · · · · · ·		,, 
Curl curl -X '' 'https: -H 'acci -H 'acci -grant_' "client '' Request UR https://s	POGY' \ //slaes.app.redkik.com/api/v2/user/osuth/token' \ emt-Type: application/json' \ tent-Type: application/json' \ type': "client_credentials", 	., t= 16		
Curl curl -X '' 'https:/ -H 'acci -H 'Com -d '{ "grant_" "client '' client }' Request UR https://s	POST' \ //slas.spp.redkik.com/spi/v2/user/cauth/taken' \ pet: application/json' \ /secret: spplication/json' \ _sd: "\$\$\$\$24Ci-d95-4865-efd4dd938682", _secret: "#369724f1e9772d195cl185ac6#f2bcd886d68637945b1b334c2372275e8a1538" L ale: spp.redkik.com/spi/v2/user/cauth/taken	, t G		
Curl curl -X '  'https:/- -H 'Com' -d '{ "grant_" "client, "client, "client, "client, Server resp. Code	POST' \ //sles.app.redkik.com/api/v2/user/oauth/token' \ pet: sapil.cation/json' \ Tent-Type: apil.cation/json' \ ypet: slist.com/api/v2/user/oauth/token 	., t: G		
Curl eur X. 1 - Nttps: - H 'acc - H 'con - d '(' "grant,' "client,' )' Server resp Code 200	POST: \ //slessapp.redkik.com/api/v2/user/osuth/token' \ /pri: apjlt.redki/jen' \ /tent-type: apjlt.redki/jen' \ /tent-type: apjlt.redki/jen' \ /tent-type: apjlt.redki/jen' \ /tent-type: apjlt.redki/scom/api/v2/user/osuth/token cerret: also724f1e9724f1e9724f1e9724f1e9724f1e9724f1e9724f1e9724f1e9724f1e972	12 (3 12) (11) (2) (3) (3) (3) (3) (3) (3) (3) (3) (3) (3) (3) (3)		
Curl euri, - X, "  "https://iece -4, euro " digrams", "claemt", "claemt", Code Code 200	Post / / // Post spprddik.com/opi//2/user/seuth/token / / ppt: spp.rddik.com/opi//2/user/seuth/token / / ppt: spp.rddik.com/opi//2/user/seuth/token / / ppt: spl.rddik.com/opi//2/user/seuth/token cutors / / / / / / / / / / / / / / / / / / /	a, tº (g)		
Curl euri-At "1 "https://i "client" "client" Code 200 Responses	<pre>/// / /// / /// lasts app.rediki.com/pi/v2/user/aasth/taken' / //  //  //  //  //  //  //  //  //</pre>	3, 12 (3) 32,7/11(2)j=4/Q101 UW4/11(2)j=4/Q101 UW4/11(2)j=4/Q101 Download		
Curl euri, - X, "  "Introps, 	<pre>PDF1 \ / / / / / / / / / / / / / / / / / /</pre>	a, ts (g)		

# Refresh Tokens

When API Tokens are used to generate an access token, they will also include a refresh token. This refresh token can be used to generate a **new** access token before the TTL of that access token expires, in order to keep that particular session alive. If the access token expires before it can be refreshed, the client\_id and secret\_id must be used again to generate a new access token.

# Utilizing the token and Listing Bookings

To GET bookings using the token:

e.g.: Authorization: Bearer <TOKEN>

For example, when using curl, one would need to include the token as a header specified on the command line:

```
$ curl -X 'GET' \
-H 'accept: application/json' \
-H 'Authorization: Bearer <TOKEN>' \
https://staging.app.redkik.com/api/v2/quote/bookings
```

#### Listing All Bookings

```
curl -X 'GET' \
    'https://staging.app.redkik.com/api/v2/quote/bookings' \
    -H 'accept: application/json' \
    -H 'Authorization: Bearer <TOKEN>'
```

### Listing a single Booking's detailed information

```
curl -X 'GET' \
```

```
'https://staging.app.redkik.com/api/v2/quote/bookings/<BOOKING_UUID> \
```

```
-H 'accept: application/json'
```

```
-H 'Authorization: Bearer <TOKEN>'
```

# Generate a Quote and Purchasing it

# Getting Setup information Before Generating a Quote

Optionally, we pre-populate a json structure with setup information around the policy context, available commodities, etc, which can help populate any UI forms upstream. This is not a required step, but it assists in setting the correct UUID's for commodities and sundry. That endpoint is here:

/quotes/setup:

🗇   🗅 Redick   App	🗙 📋 Redikik   App	× () Swagger UI × +					-	ø ×
← C 🗅 https://sale	is app.redkik.com/api/v2/apidoc/?urls.prim	aryName=QuoteService#/QuoteController/quotesSetup		A <sup>%</sup>	ίō	r: @	ø	e- 🕄
	Quotecontroller							* Q
	CET /api/v2/quote/quote	s/setup		^ ≜				~.
	Parameters		Cance					٩
	No parameters							ø
								+
		Execute	Clear					
	Responses							- 10
	Curl							
	curl -X 'GET' \ 'https://sales.app.redkik.com/api	/v2/quote/quotes/setup" \						
	-H 'accept: application/json' \ -H 'Authorization: Bearer eyThbGc	iOIJTU/IINIIsIaR5cIIGIkpNVCJ9.eyJleNAiOjE2N/EJNTV2MjAsIaSIZIIGNTV3M	TU MOTSACALAAFAT jaak jeant vah jean Caqata joi et indy ant zaki 117 jaal tozhe e tot ave toant rake 1142 e eziz Tu motsacalaafat jaak jeant vah jean Caqata joi et indy ant zaki 117 jaak e tot ave toant rake 1142 e eziz jaak					
	Request URL			~				
	https://sales.app.redkik.com/api/v2	//quote/quotes/setup						
	Server response							
	Code Details							
	200 Response body							
	<pre>{     "commodities": [     { </pre>			Û.				
	"id": "b4eBef5c- "nome": "Artisti "createdot": "10	cf12-4001-0f0c-135142217187", C T≤ WerkT, D-10-115144-16-12-2762".						
	"deletedAt"; mull							
	f "id": "d50645a7-a "name": "Automobi	4400-457a-9f38-bafab6ed38b1", lles",						
	"createdAt"  "28 "deletedAt"  mull	22-19-11T14:18:15.7142", I						
	{ "1d"  "40001108-	929-4f98-acfd-4c74cf421775*,						
	"createdAt"  "28 "deletedAt"  null	HE & SQUFITS", 22-10-11714:13:51.8062",						
	}, { "id"  "hebidatis-"	38cb-45ef-9378-e9564db426e7*。						œ
	"name": "Bitcoin "createdAt": "200	Server", 22-10-11T14:13:32.0482",						- 🕸

# Creating a Quote

To create a quote, one must POST the following JSON structure to the /Bookings/quote endpoint (this is in its most basic form):

#### /quote/quotes/quote:

E.g.:



# Purchasing the Quote

To purchase the quote, one retains the ID (64-bit UUID) returned with the quote (**in bold**) and submits to the <u>/purchase</u> endpoint.



# Cancelling the Quote

To cancel a booking, use the /cancel endpoint

# Advanced Quoting

# Quoting from multiple policies

# **Finding Policy IDs**

Some organizations have multiple policies from which they may quote. In those instances, the policies are listed in the /setup call.



# Quoting on behalf of

To create a booking on behalf of a child org, simply pass that child org's id with the quote:

"orgId": "cf417c93-8486-4c28-89c6-effa05987e67"

# Obtaining the Certificate of Insurance TBD

Referrals TBD

Referring a Booking Listing Referred Bookings Accepting a Referred Booking (Underwriter) Purchasing a Referred Booking (Customer)

#### **API Swimlane**

