

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](#)

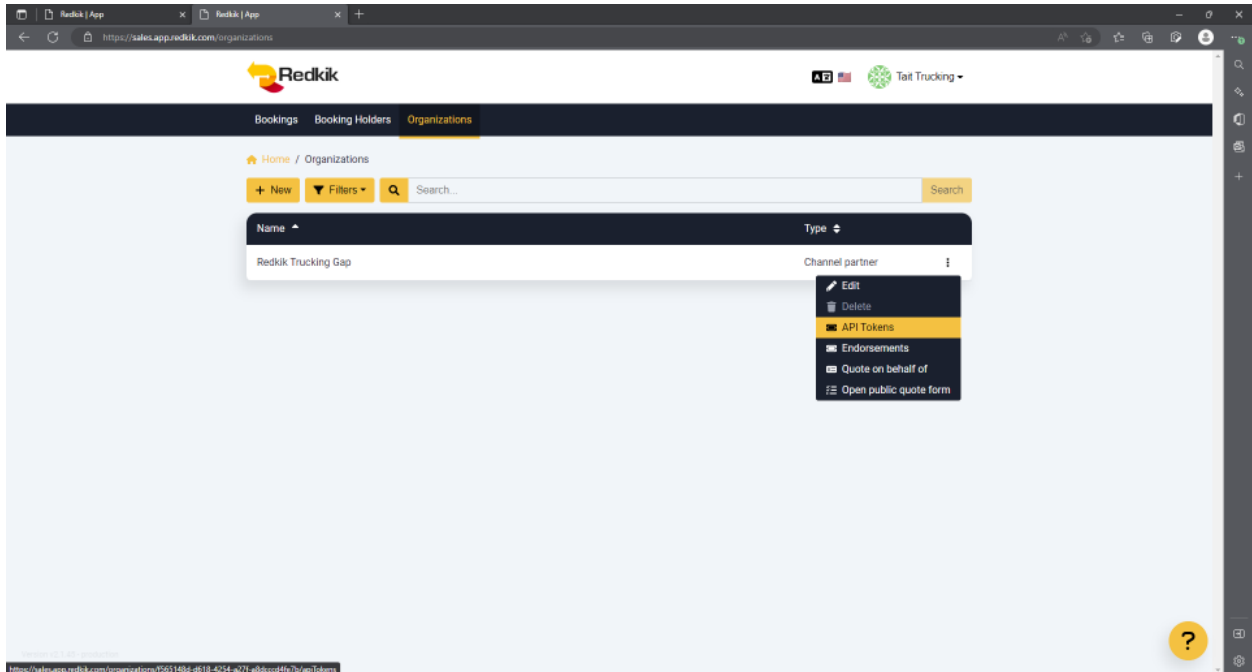
The authentication flow is as follows:

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

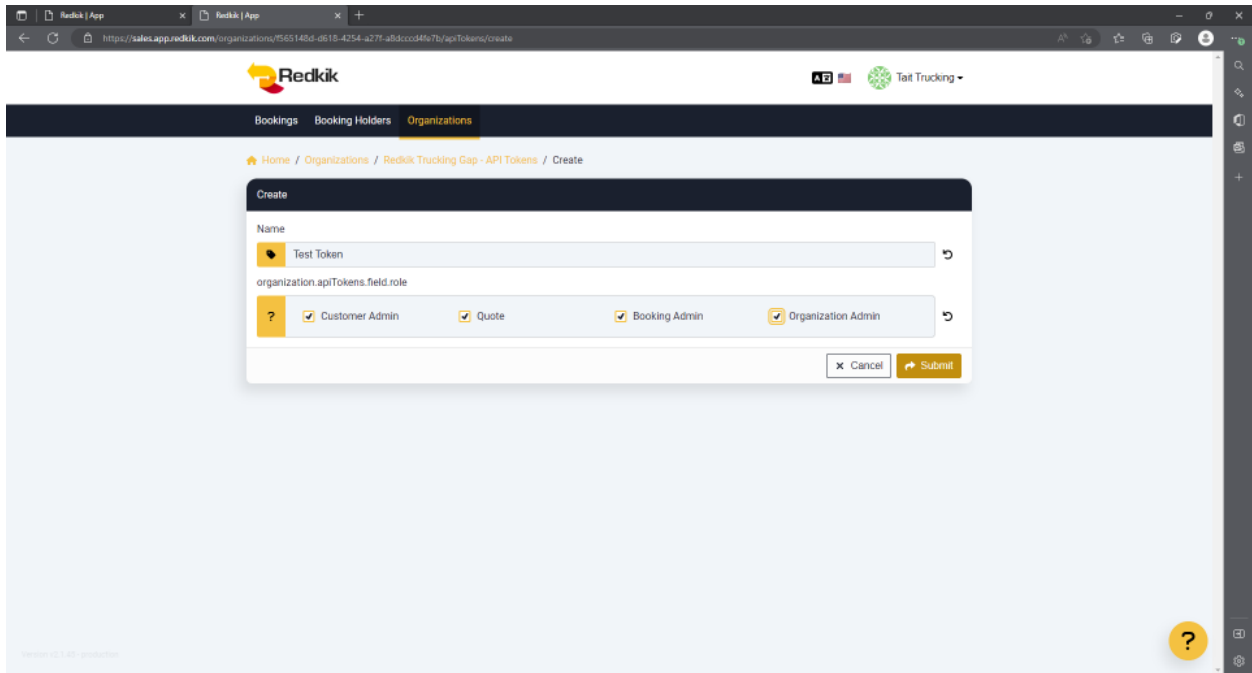
Generate API Tokens

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.

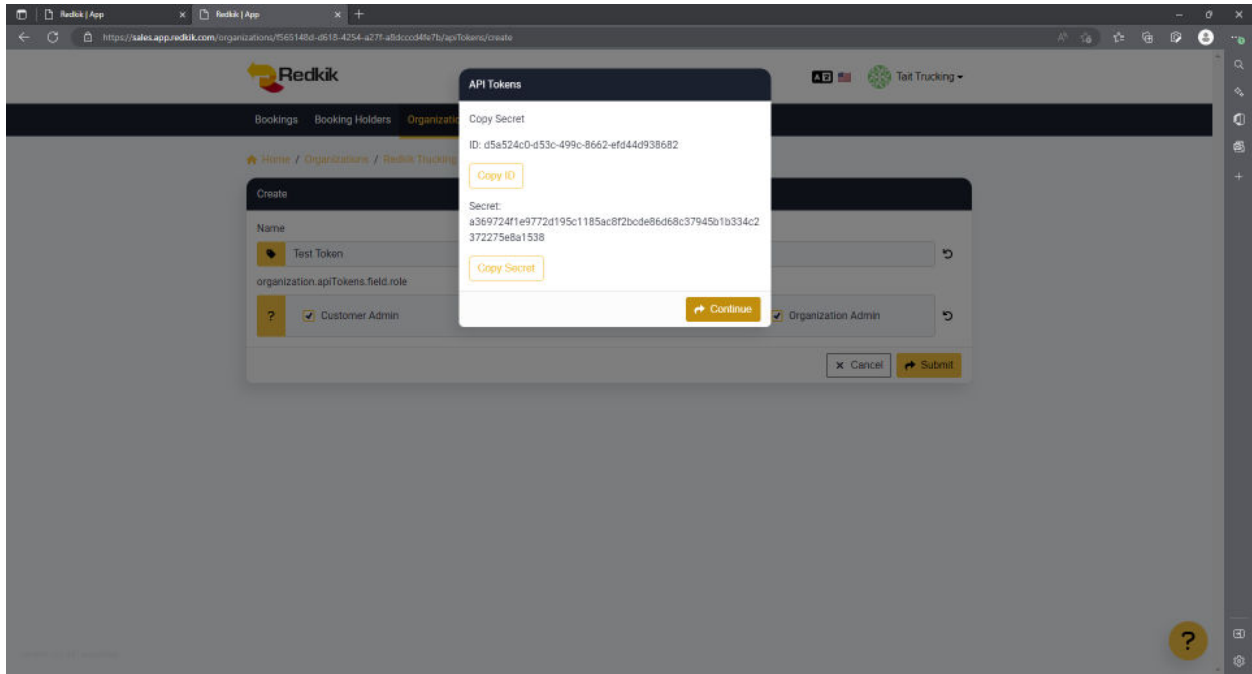
Browse to [Organizations](#), select your organization (or a child-org if creating a Quote "On Behalf Of" that child-org), right click on the org's config button, and select API Tokens:



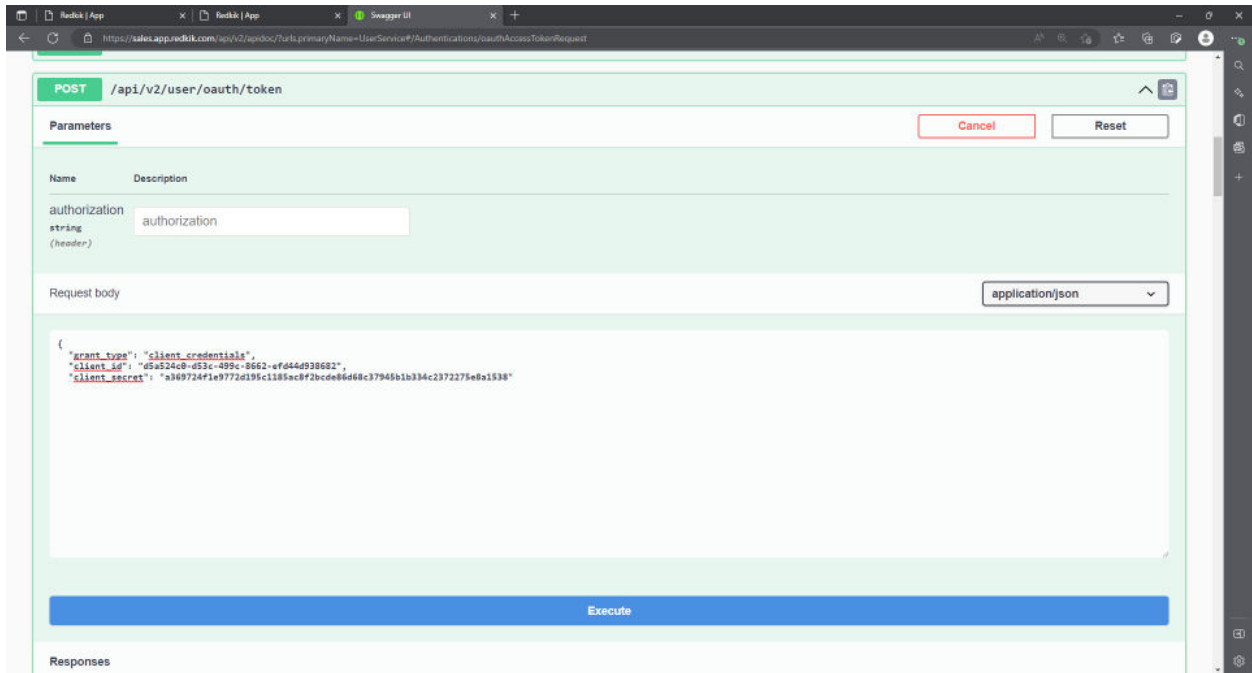
Add a new API Token:



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 [/oauth/token](#) endpoint:



Producing an Access token and Refresh 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

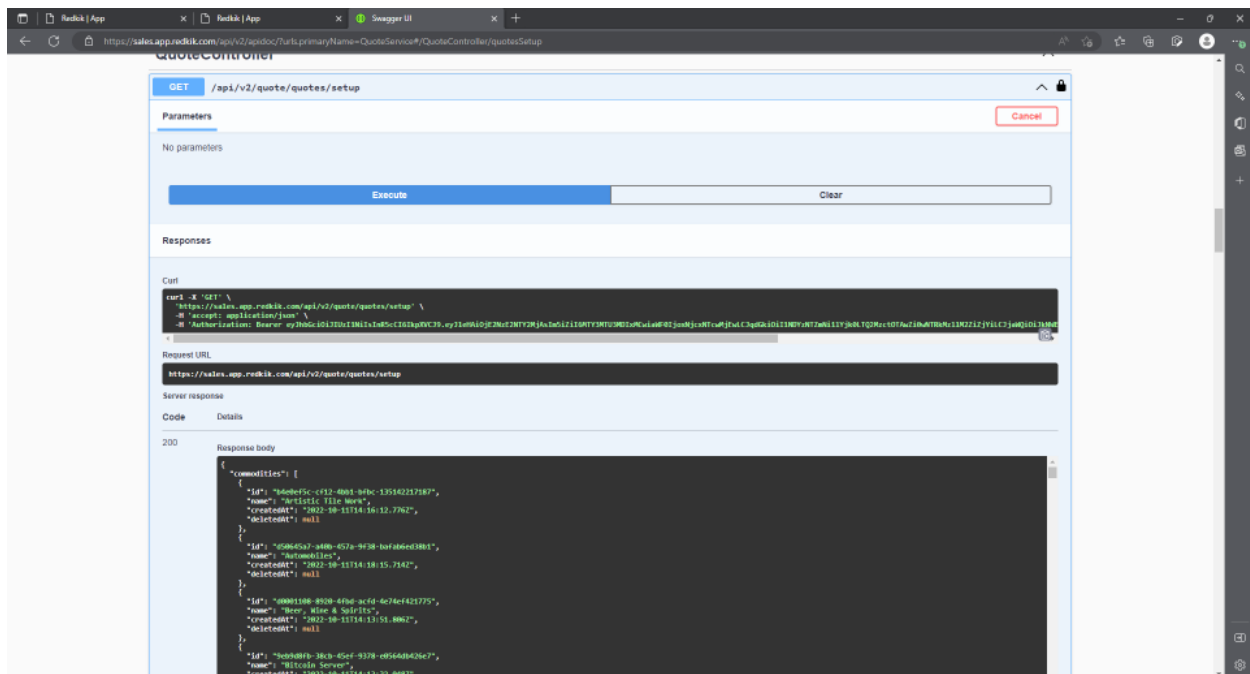
```
curl -X 'GET' \  
'https://staging.app.redkik.com/api/v2/quote/bookings/<BOOKING_UUID>' \  
-H 'accept: application/json' \  
-H 'Authorization: Bearer <TOKEN>'
```

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>:



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.:

```
curl -X 'POST' \
  'https://sales.app.redkik.com/api/v2/quote/quotes/quote' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{
isPublic: false,
commodities: [
{
commodityId: "d9a727f0-4447-4646-8b9f-e567d7e0d174",
insuredValue: 100,
currencyId: "c14fc724-f875-468b-a57b-77290e66e787",
},
],
insuredValue: 1000,
originFormatted: "Länsikatu 15, FI-80110 JOENSUU, FINLAND",
destinationFormatted: "Länsikatu 15, FI-70820 KUOPIO, FINLAND",
startDate: "2022-10-23T03:00:00.000+03:00",
endDate: "2022-10-24T03:00:00.000+03:00",
transportType: 1,
customerId: "9e554527-4a04-40b5-a297-7d65a1f47c59",
}'
```

Purchasing the Quote

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

```
curl -X 'POST' \
  'https://staging.app.redkik.com/v1/quote/bookings/purchase' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <TOKEN>' \
  -d '{
    "offerId": "037e756d-c902-4dac-86c3-7609a542235c"
  }'
```

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.

```
"policies": [
  {
    "id": "c4b94d1c-961a-4099-88c8-aef19b9b881d",
    "alias": "Cargo Policy"
  },
  {
    "id": "cf417c93-8486-4c28-89c6-effa05987e67",
    "alias": "Warehouse Policy"
  }
]
```

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

