

# GOSHIP



**LTL Quote API  
Integration Documentation**

# Introduction

Integrating with GoShip’s LTL Quote APIs can provide small businesses with numerous benefits, including a reduction in operational costs and overhead. GoShip’s APIs automate many of the manual processes involved in quoting and booking shipments. As a result, businesses can save time, money, and resources, which can be used for other areas of the business. Furthermore, GoShip’s APIs can also improve the overall customer experience, by providing real-time shipping information, which is essential for small businesses looking to grow and expand.

## Table of Contents

- [Authentication](#) ..... 3
- [Generating the API Token / Key](#) ..... 3
- [Endpoints and Requests](#)..... 5
- [Using the Document Explorer](#) ..... 7
- [Integration using NodeJS](#)..... 9
- [Getting the Rates \(Explorer\)](#) ..... 10
- [Preparing the Order](#)..... 11
- [Finalizing a Shipment Booking via Vendor Account](#) ..... 12

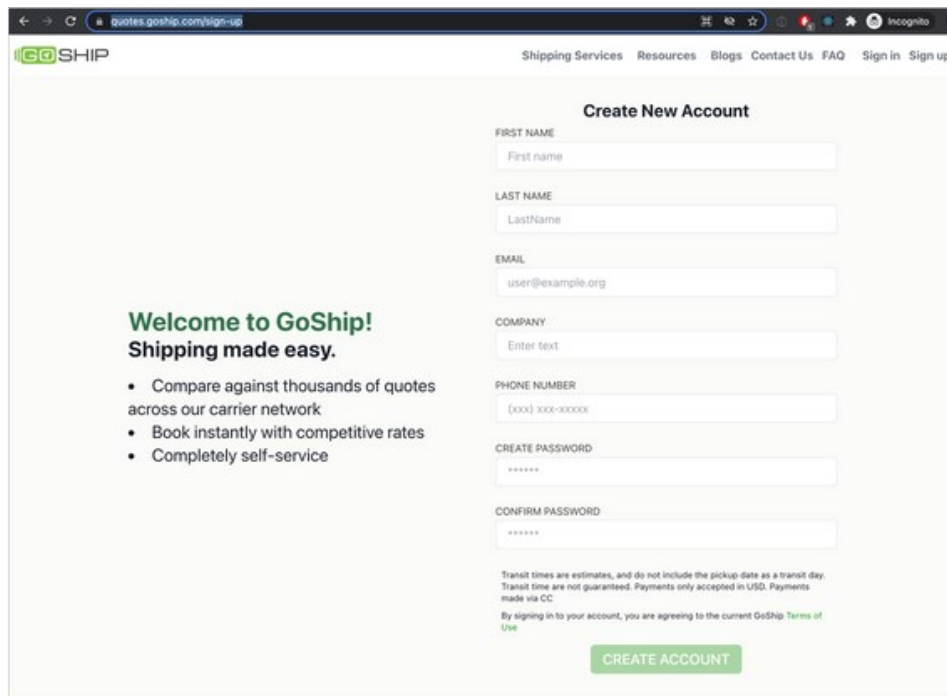
# Authentication

All GraphQL queries require a valid access key. Include your key as an **X-GoShip-API-Key** header on all queries. Example screenshots are provided further down in this document.

## Generating the API Token / Key

In order to generate an access key you must have a GoShip account.

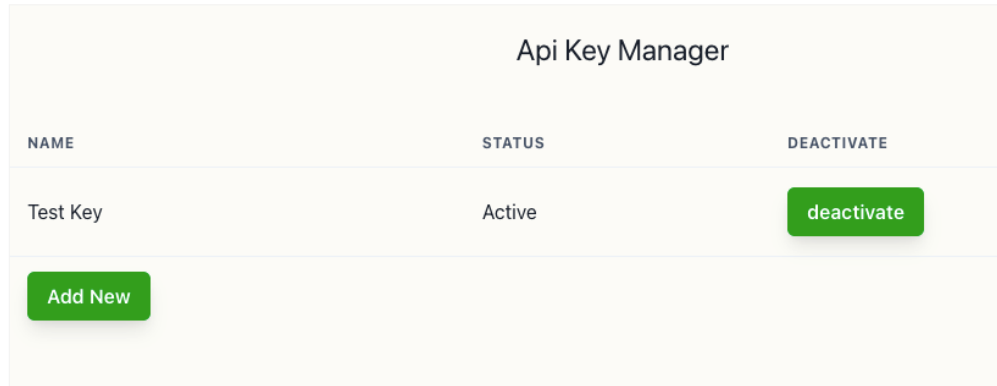
1. To create an account (If you don't already have one)
  - a. Visit [quotes.goship.com/sign-up](https://quotes.goship.com/sign-up)



- b. Enter your account details
- c. Click 'Create an Account'
- d. Once an account has been created you will receive a verification email at the email address you specified during account creation
- e. After verifying your email and logging in, you will be able to progress to the next step (Generate your API Key)

## 2. Generate your API key

- a. Once logged in, navigate to [quotes.goship.com/apikeys](https://quotes.goship.com/apikeys)

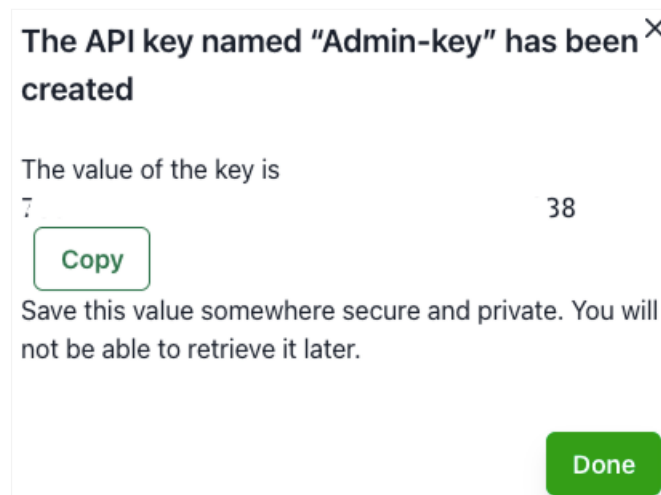


- b. Click the 'Add New' button to add a new API key.

The screenshot shows a 'Create new API key' dialog box. It has a title bar with a close button (X). Below the title is a 'Name' label and a text input field containing 'Admin Key'. At the bottom of the dialog are two buttons: a green 'Create' button and a red 'Cancel' button.

- c. Enter a name for your API Key and then click the 'Create' button. The system will automatically generate your secure key. **Please make sure to copy the secure key, and keep it safe.**

**⚠️ Treat the API Key like a password ⚠️**

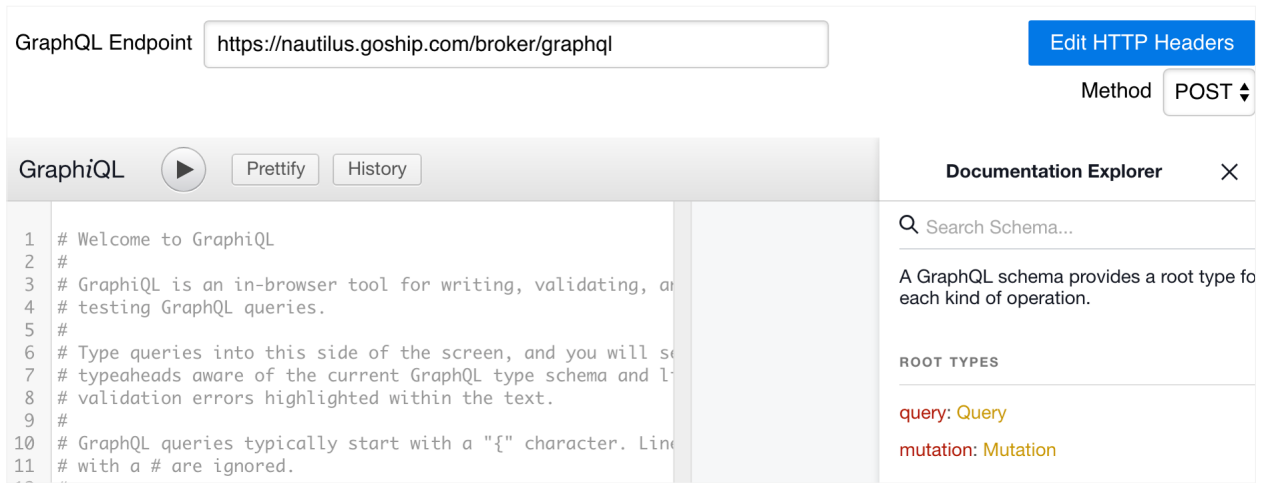


# Endpoints and Requests

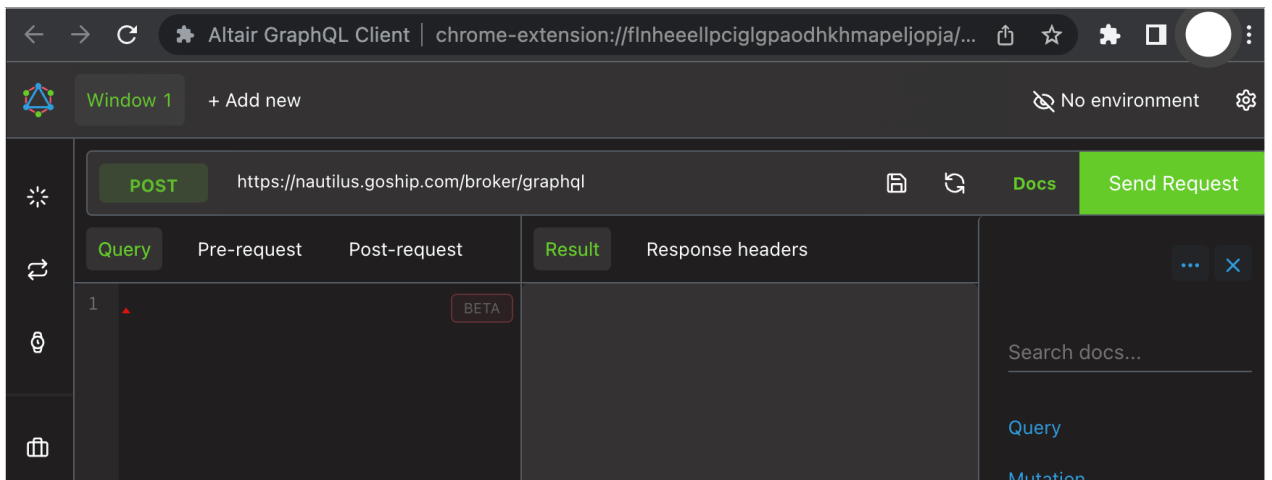
All GraphQL endpoints are handled from this URL: <https://nautilus.goship.com/broker/graphql>

To test a request and authentication key, open up a GraphQL explorer. For example, we have two explorers as an example.

## 1. GraphiQL (preferred)



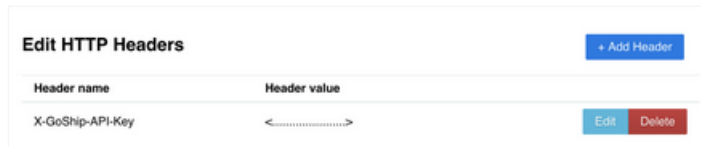
## 2. Altair (Google Chrome Extension)



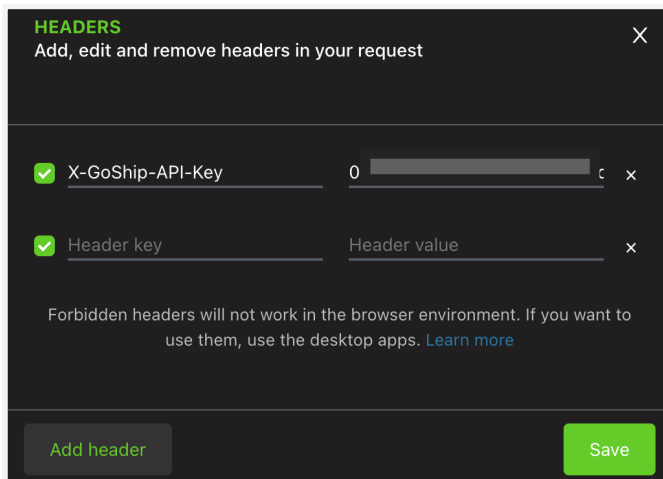
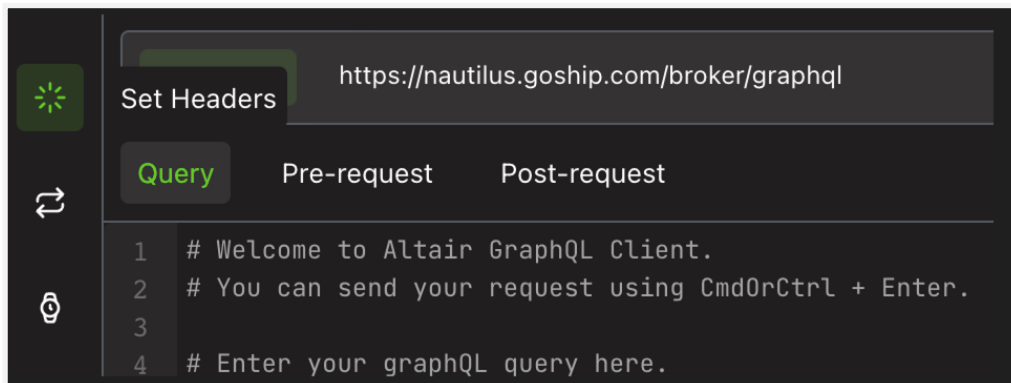
Point the GraphQL explorer to the URL <https://nautilus.goship.com/broker/graphql> and set the header for authentication

1. Name: **X-GoShip-API-Key**
2. Value: **<secret key copied from API keys page—created in Step 2>**

### GraphiQL



### Altair



After saving the header, you can explore the broker API.

# Using the Document Explorer

For elaborate requests and responses, both GraphiQL and Altair have a Documentation Explorer section where all the queries and mutations can be viewed along with their request and response parameters.

## 1. GraphiQL

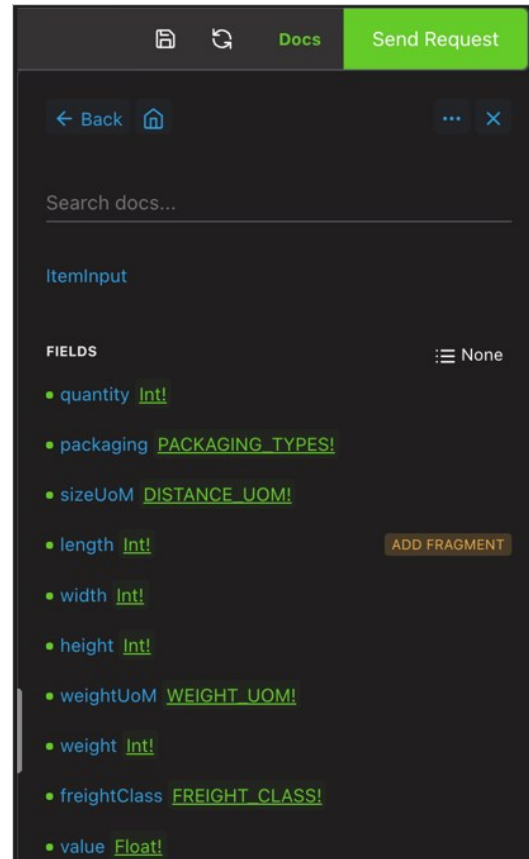
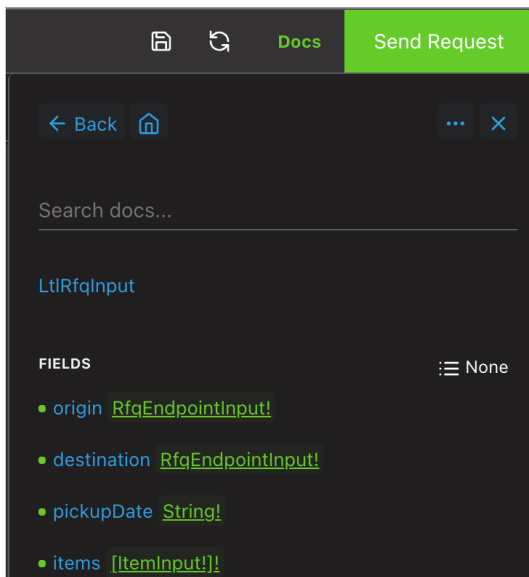
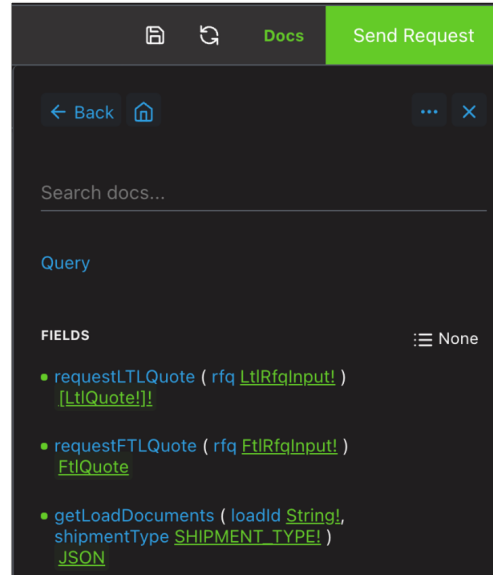
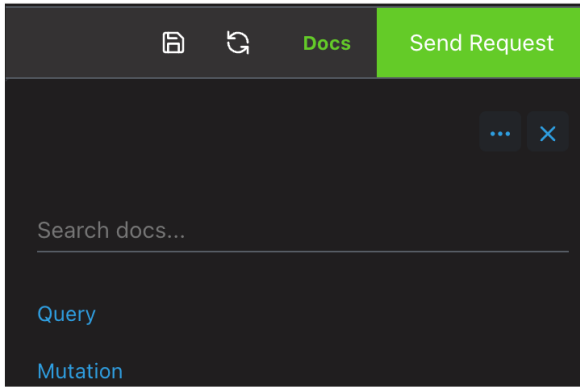
On the right side of the explorer, there is a section with the heading 'Documentation Explorer' which contains queries and mutations along with request and response parameters

The image displays four screenshots of the GraphiQL Documentation Explorer interface, arranged in a 2x2 grid. Each screenshot shows a different level of detail in the documentation.

- Top Left:** The main 'Documentation Explorer' view. It features a search bar labeled 'Search Schema...', a description: 'A GraphQL schema provides a root type for each kind of operation.', and a section for 'ROOT TYPES' listing 'query: Query' and 'mutation: Mutation'.
- Top Right:** A view for a specific query, 'requestLTLQuote'. It shows a search bar 'Search Query...', 'No Description', and a 'FIELDS' section listing: 'requestLTLQuote(rfq: LtlRfqInput!): [LtlQuote', 'requestFTLQuote(rfq: FtlRfqInput!): FtlQuote', and 'getLoadDocuments(loadId: String!, shipmentType: SHIPMENT\_TYPE!): JSON'.
- Bottom Left:** A view for the 'LtlRfqInput' type. It shows a search bar 'Search LtlRfqInput...', 'No Description', and a 'FIELDS' section listing: 'origin: RfqEndpointInput!', 'destination: RfqEndpointInput!', 'pickupDate: String!', and 'items: [ItemInput!]!'.
- Bottom Right:** A view for the 'ItemInput' type. It shows a search bar 'Search ItemInput...', 'No Description', and a 'FIELDS' section listing: 'quantity: Int!', 'packaging: PACKAGING\_TYPES!', 'sizeUoM: DISTANCE\_UOM!', 'length: Int!', 'width: Int!', 'height: Int!', 'weightUoM: WEIGHT\_UOM!', 'weight: Int!', and 'freightClass: FREIGHT\_CLASS!'.

## 2. Altair

Similar to GraphQL, document explorer can be viewed on the right panel. If it is not visible there is a 'Docs' button to the left of 'Send Request' that will toggle the document explorer



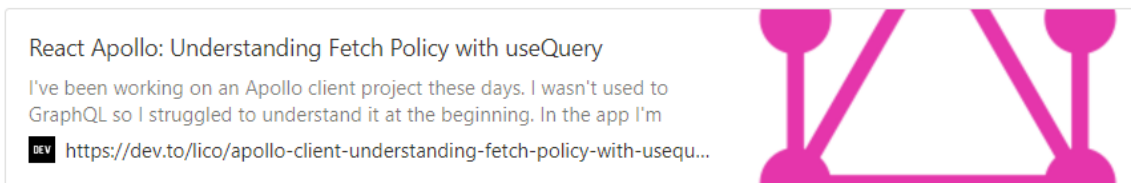


# Integration using NodeJS

## 1. Library

- a. apollo ^2.33.4
- b. graphql ^15.5.1

## 2. Reference: (<https://dev.to/lico/apollo-client-understanding-fetch-policy-with-usequery-1gfa>)



## 3. Setting up Apollo Client

```
const httpLink = createHttpLink({
  uri: GRAPHQL_ENDPOINT,
});

const authLink = setContext(async (_, { headers }) => {
  return {
    headers: {
      ...headers,
      'X-GoShip-API-Key': '<secret key copied from API keys page>',
    },
  };
});

export default new ApolloClient({
  link: authLink.concat(httpLink),
  cache: new InMemoryCache(),
});
```

## Getting the Rates (Explorer)

To request shipping rates, open the explorer and pass the required information in the query field. An example query is below. Query parameters can be changed to see different results.

```
query{
  requestLTLQuote(rfq:{
    origin:{
      postalCode:"16066"
      addressType: RESIDENTIAL
    }
    destination:{
      postalCode:"16066"
      addressType: BUSINESS_W_DOCK
    }
    pickupDate:"2022-11-12" //this must be a future date
    items:{
      quantity:1
      packaging: PALLET
      sizeUoM:in
      length:10
      width:10
      height:10
      weightUoM:lbs
      weight:200
      freightClass:CLASS_50
      value:100
      itemCondition:NEW
      stackable:false
      hazardous:false
      description:"rest"
      pieces:1
      country:USA
    }
  }){
    cost
    carrier{
      name
      id
    }
    id
    deliveryDate
  }
}
```

## Preparing the Order

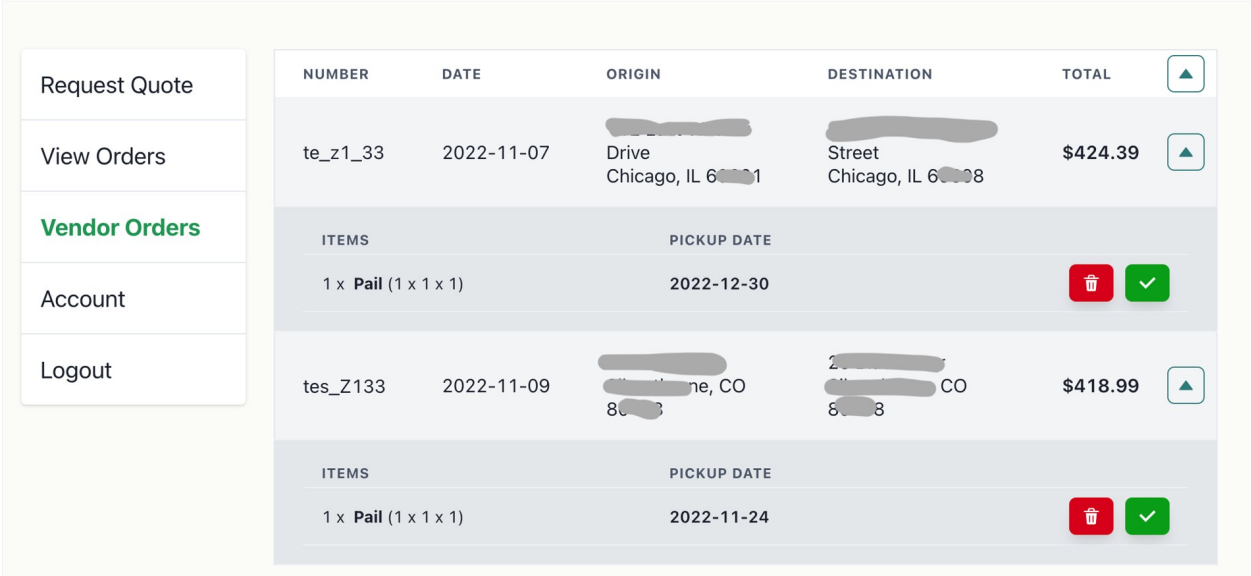
When checkout is done by the customer after selecting a quote, the request needs to be sent as a mutation to store the shipment order. The mutation only stores the quote information and it **does not actually process a shipment. A shipment must be confirmed and paid for by the vendor using the GoShip Vendor Orders UI. The steps and screenshots are below.**

### Mutation

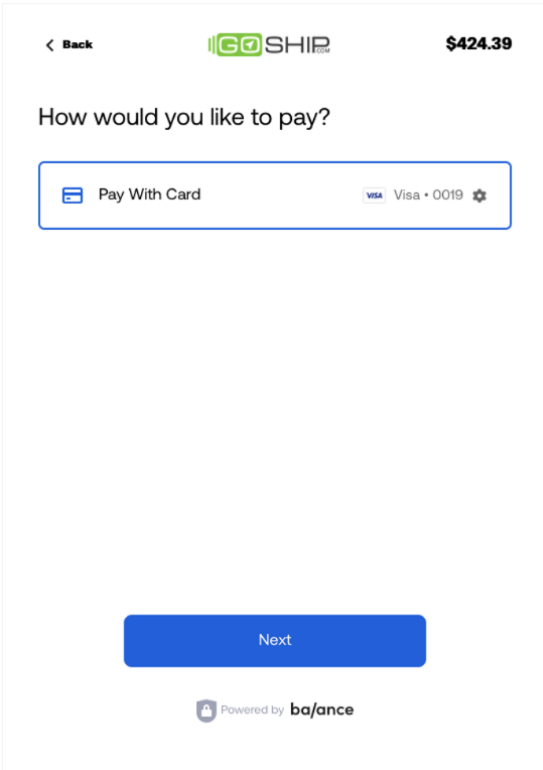
```
mutation{
  prepareOrder(input:{
    quoteId:"283cdf9e-2e52-439b-b066-1eb4f594387e" //this is the id returned from the
    orderNumber:"tes_Z133"
    origin:{
      address:{
        streetAddress:"1234 Main Street"
        postalCode:"12345"
        city:"Schenectady"
        state:"NY"
        country:USA
        firstName:"John"
        lastName:"Doe"
        companyName:"JohnDoeCompany"
        phone:{
          areaCode:"888"
          countryCode:"1"
          number:"1234567"
        }
        email:"john@doe.com"
      }
      startTime:"2023-11-12 10:53:00.888 -0800"
      endTime:"2023-11-12 10:53:00.888 -0800"
    }
    destination:{
      address:{
        streetAddress:"1234 Main Street"
        postalCode:"12345"
        city:"Schenectady"
        state:"NY"
        country:USA
        firstName:"John"
        lastName:"Doe"
        companyName:"JohnDoeCompany"
        phone:{
          areaCode:"888"
          countryCode:"1"
          number:"1234567"
        }
        email:"john@doe.com"
      }
      startTime:"2023-12-12 10:53:00.888 -0800"
      endTime:"2023-12-12 10:53:00.888 -0800"
    }
  }){
    vendorOrder{
      id
    }
    error
  }
}
```

# Finalizing a Shipment Booking via Vendor Account

- 1. Navigate to <https://quotes.goship.com/booking/vendor-orders>, this page is also accessible from the GoShip Shipping Dashboard.



- 2. After an order has been placed and the mutation call for prepareOrder returns a successful response, the order can be seen in the vendor orders table. Payment can be made using the green 'checkmark' button. After the button is clicked, a payment modal will open up. Enter the credit card details and submit the payment.



## 4. Querying the Rates

```
const GET_RATES = gql`
  query RequestLTLQuote($rfq: RFQ!) {
    requestLTLQuote(rfq: $rfq) {
      cost
      carrier{
        name
        id
      }
      id
      deliveryDate
    }
  }
`;

export default async function requestLTLQuote(rfq: RFQ) {
  const resp = await graphqlClient.query<RequestLTLQuote>({
    query: GET_RATES,
    variables: { rfq },
  });

  const rates = resp?.data?.requestLTLQuote;
  return { rates };
}
```

In the case where payment is being submitted after the estimated ship date, an expired pickup date prompt will be displayed. Enter a new future pickup date. **It must be a future pickup date to book the shipment.**

### Pickup Date ×

Your shipment pickup date is no longer valid.  
Please select a new date before payment.

01/05/2023

Book Load

3. After a successful payment, a shipment booking confirmation email is sent to the email address used to create the vendor account. The email contains the BOL document and details of the shipment.
4. Vendors can track and view shipment documents (e.g. BOL and Invoices) for all shipments from their GoShip View Orders Dashboard.

Shipping Dashboard Resources

### My Order History

✦ Earn \$100 in GoShip credit! Get Started

Search by BOL

	STATUS	BOL	TYPE	CARRIER	TRACKING #	PICKUP	DELIVERY
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Request Quote</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">View Orders</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Vendor Orders</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Account</div> <div style="border: 1px solid #ccc; padding: 5px;">Logout</div>	<span style="font-size: 24px; font-weight: bold; color: #008000;">ABF</span>	1-██████-3	LTL <span style="font-size: 18px;">↓</span>	<span style="font-size: 18px; color: #008000;">\$</span> ABF FREIGHT SYSTEM, INC.		Cranberry Township, PA Oct 5	Cranberry Township, PA Oct 6
	<span style="font-size: 24px; font-weight: bold; color: #008000;">ABF</span>	1-██████-6	LTL <span style="font-size: 18px;">↓</span>	<span style="font-size: 18px; color: #008000;">\$</span> ABF FREIGHT SYSTEM, INC.		Cranberry Township, PA Sep 29	Cranberry Township, PA Sep 30
	<span style="font-size: 24px; font-weight: bold; color: #008000;">ABF</span>	1-██████-1	LTL <span style="font-size: 18px;">↓</span>	<span style="font-size: 18px; color: #008000;">\$</span> ABF FREIGHT SYSTEM, INC.		Cranberry Township, PA Sep 16	Cranberry Township, PA Sep 19
		1407-██████-0	LTL <span style="font-size: 18px;">↓</span>	<span style="font-size: 18px; color: #008000;">\$</span> OLD DOMINION FREIGHT LINE INC		Cranberry Township, PA Aug 5	Silverthorne, CO Aug 12