

# API Partner Integration Documentation



# Table of Contents

[Overview](#)

[Endpoints](#)

[Workflow](#)

[Additional quote and price indication details](#)

[Critical data points for accurate pricing](#)

[Authentication](#)

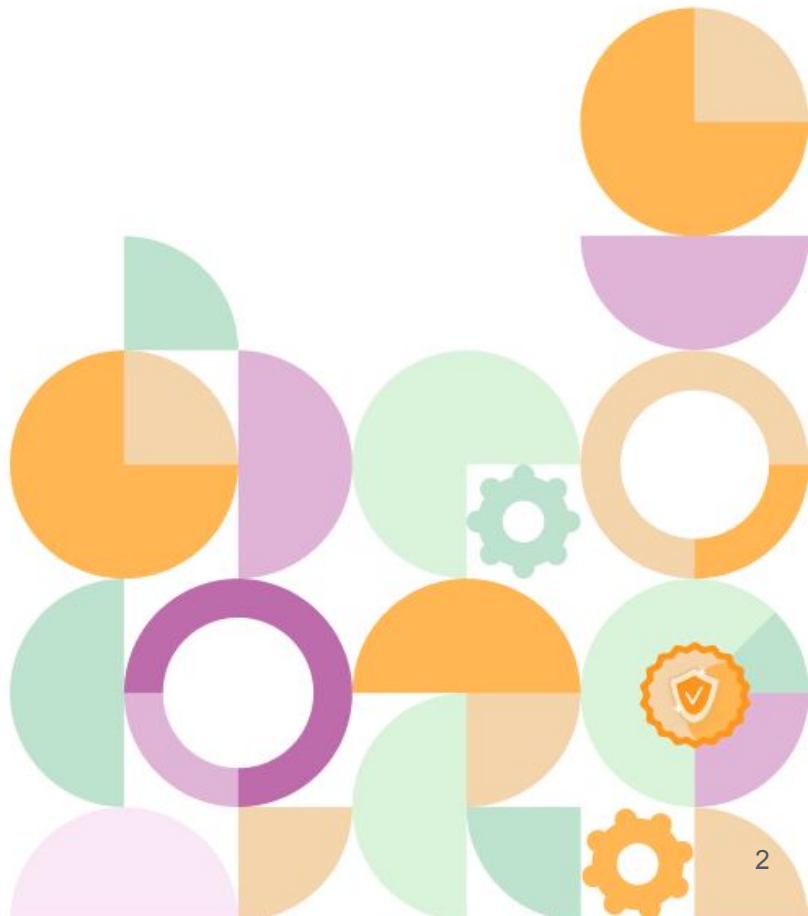
[Responses](#)

[Detailed responses - quote endpoint](#)

[Manual API testing with Swagger](#)

[Developing your API using Swagger](#)

[FAQs](#)



# Overview

At Pie, our goal is to make workers' comp insurance easy for our partners and their small business clients.

By integrating with our API, you can:

- Check our appetite for coverage
- Get a price indication
- Request a quote on behalf of your client
- Upload documents with your submission for Pie's underwriting review
- Retrieve bindable quotes in PDF format to share with your client

You can call our RESTful individual endpoints (using JSON) to perform the following actions:

- GET - retrieve specific information, which may include arrays and sets of data and information
- POST - create/add a new API element

Our [Swagger site](#) provides full documentation on supported input objects, values, requirements, and formats. This document covers details on integrating and testing with our API, and provides details around each endpoint.

During your development and testing, we will provide you with credentials specific to the Pie Post Prod environment: <https://api.post-prod.pieinsurance.com/>. Once logged into the portal you can access the API keys. All testing will be done via the Post Prod environment at <https://api.post-prod.pieinsurance.com/api/v1/Quotes>.

When you are ready to launch, Pie will require a certification of your application. Upon completion of the certification process, we will supply credentials for the Pie production environment: <https://api.pieinsurance.com/api/v1/Quotes>.

We recommend using [Postman](#) for testing our API in the Post Prod environment. We will work with you to validate tests you submit in our Post Prod environment to certify your application for use in our production environment.

If you have any questions or issues, please contact your Pie representative. We have dedicated staff to support your integration and are happy to work with you to answer any questions.



# Endpoints

We have a variety of endpoints you can use in the way that best matches your user application and experience.

Endpoint	URL	Action	Description
Token	<a href="https://api.post-prod.pieinsurance.com/oauth2/token">https://api.post-prod.pieinsurance.com/oauth2/token</a>	POST	Accepts credentials and returns the JWT authentication token.
Class Code Search	<a href="https://api.post-prod.pieinsurance.com/api/v1/ClassCodeSearch">https://api.post-prod.pieinsurance.com/api/v1/ClassCodeSearch</a>	POST	Returns possible matching class codes based on a state and description provided
Appetite	<a href="https://api.post-prod.pieinsurance.com/api/v1/Appetite">https://api.post-prod.pieinsurance.com/api/v1/Appetite</a>	POST	Returns class codes for which Pie has an appetite to write based on state (or ZIP) provided
Price Indication	<a href="https://api.post-prod.pieinsurance.com/api/v1/PriceIndication">https://api.post-prod.pieinsurance.com/api/v1/PriceIndication</a>	POST	Returns a price indication based on a minimal set of customer data points
Eligibility	<a href="https://api.post-prod.pieinsurance.com/api/v1/Eligibility/questions">https://api.post-prod.pieinsurance.com/api/v1/Eligibility/questions</a>	POST	Returns eligibility questions based on class code and state provided. The set of questions and responses can be sent in the quote endpoint for the most accurate price and status indication.
Quotes	<a href="https://api.postprod.pieinsurance.com/api/v1/Quotes">https://api.postprod.pieinsurance.com/api/v1/Quotes</a>	POST	Accepts critical data points needed to return a status and quote if applicable. In addition, accepts optional data points that can be used later in bind or policy issue.
Quote Document	<a href="https://api.post-prod.pieinsurance.com/api/v1/QuoteDocument">https://api.post-prod.pieinsurance.com/api/v1/QuoteDocument</a>	POST	Allows the ability to append files to a specific quote (based on QuoteID)
Quote PDF	<a href="https://api.post-prod.pieinsurance.com/api/v1/QuotePdf">https://api.post-prod.pieinsurance.com/api/v1/QuotePdf</a>	GET	Accepts a quote id and returns a PDF for <b>bindable</b> quotes



# Workflow

Based on the available endpoints, you can configure the workflow in the way that best matches your user application and experience.

First, verify that the Pie Quote API is available by attempting to get a token.

Your application can capture a search term from your end user, and call the **Class Code Search** endpoint to find matching class codes. In addition, **Class Code Search** can attempt to infer a class code from the business name, which can be useful if your application already has a name (e.g., ABC Plumbing will return plumbing related class codes for the applicable state as possible matches).

Your application can call the **Appetite** endpoint to determine which class codes and states Pie supports early in your workflow. You can use this endpoint improve your user experience by avoiding requesting a Quote from Pie where we have no appetite.

If you would like to provide your user with the ability to get a preliminary price before deciding to move forward with filling out a full quote, you can collect a minimal set of data points and submit a request to the **Price Indication** endpoint.

Prior to calling the Quote endpoint, your application should call the **Eligibility** endpoint to retrieve questions required in the quote response. The questions are grouped into a class code/state combination.

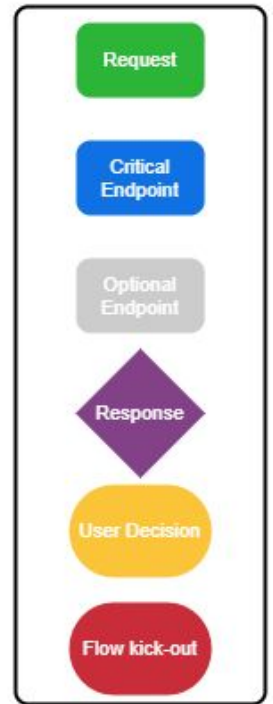
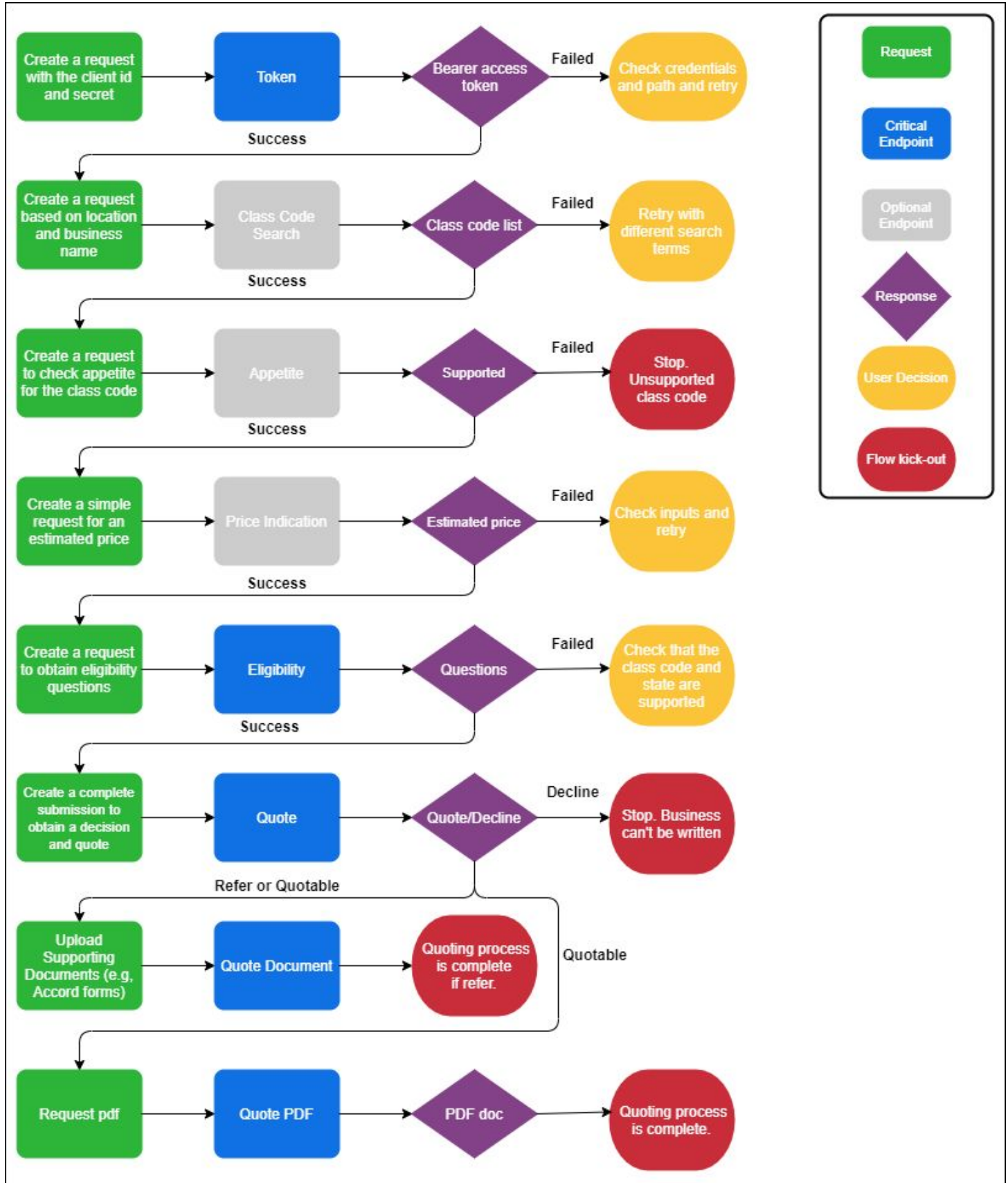
Once the required information is collected from the end-user, your application can create a request for the **Quote** endpoint. It's important to supply accurate and complete information to ensure the most accurate response. The Quote endpoint will supply a response (more on responses below), including an identifier for the quote, which you can then use to pass supporting documentation (e.g., ACORD 130 forms) to the **Quote Document** endpoint.

Note: Pie's underwriting team does not require supporting documentation to review referred submissions, however providing supporting documentation can speed up the underwriting decisioning process and is strongly recommended.

If your **Quote request** returns a status of **Quotable**, your application can retrieve a PDF via the **Quote PDF** endpoint using the quote id from the **Quote** response.



# Workflow cont'd



# Additional quote and price indication details

Getting an accurate quote or price indication is dependent on sending full and accurate data within these endpoints.

The difference between the two calls is as follows:

- **Price Indication** requires the minimum set of data to return an estimated price.
- **Quote** collects the full set of data, including eligibility questions, that can impact price and quote status (decline / refer). Many of the additional details collected in Quote are used in setting up the fuller set of data needed for bind and in underwriting review, if needed.

To get prices that are aligned between the two calls when calling both in your flow, it is critical that you enter the same underlying data. Data sent in Price Indication does need to also be sent in Quote, if using both endpoints in your workflow.

**In order to get accurate prices, it is important that we get accurate data for the following:**

- Correct class code and payroll amounts
- Status of current coverage - understanding whether the client is covered currently, a new venture without prior coverage, or has had prior coverage but is currently in a lapsed state
- Length of prior continuous coverage and whether there were any historical lapses
- Number of claims incurred during previous coverage period and the coverage period in which they occurred

When you share prior carrier claims information (in both Price Indication and Quote endpoints), we interpret each set of (repeatable) records in the array to cover 1 year of history. In quote, we have a Prior Carriers array that we expect you will repeat for each year of previous coverage, starting with the most recent history (i.e. we expect that the first record will be your first expiring term).

**Note:** The pricing and decisioning for new ventures and current lapses are different, so it is critical that this information is passed through correctly to our system. The year in which claims were incurred can also impact the pricing and decisioning.



# Critical data points for accurate pricing

Data Description	Quote Endpoint	Price Indication Endpoint
<p>Understanding the state of current coverage for a customer:</p> <ul style="list-style-type: none"> <li>Are they currently covered, and if not, why?</li> </ul>	<ul style="list-style-type: none"> <li>currently Covered</li> <li>not Currently Covered Reason               <ul style="list-style-type: none"> <li>New business</li> <li>NonPayment</li> <li>Audit NonCompliance</li> <li>No employees</li> <li>Other</li> </ul> </li> <li>not Currently Covered Description</li> </ul>	<ul style="list-style-type: none"> <li>currently Covered</li> <li>not Currently Covered Reason               <ul style="list-style-type: none"> <li>New business</li> <li>NonPayment</li> <li>Audit NonCompliance</li> <li>No employees</li> <li>Other</li> </ul> </li> <li>not Currently Covered Description</li> </ul>
<p>Understanding the coverage history:</p> <ul style="list-style-type: none"> <li>Duration of prior coverage</li> <li>Claim counts and the years in which they occurred</li> <li>Details of any years where there was a lapse in coverage</li> </ul> <p><b>Note:</b> We will interpret each occurrence of prior carrier data as 1 year of historical information. We will interpret the first record to be the most recent, expiring term.</p> <p>In Price Indication, we will interpret each claimcount year as data for 1 year of historical information, starting with the most recent, expiring term.</p>	<p><b>Repeat this object for each year of prior coverage.</b> Note lapses and claims accurately in the year in which they occurred.</p> <p>We expect this object to be blank if the customer is a new venture with no prior coverage history.</p> <p><b>Prior Carriers</b></p> <ul style="list-style-type: none"> <li>name</li> <li>effective Date</li> <li>expiration Date</li> <li>policy Number</li> <li>modification Factor</li> <li>total Premium</li> <li>nonZero Claim Count</li> <li>zeroDollar Closed Claim Amount</li> <li>lapse Period</li> <li>lapse Reason           <ul style="list-style-type: none"> <li>NonPayment</li> <li>Audit NonCompliance</li> <li>No employees</li> <li>Other</li> </ul> </li> <li>lapse Reason Description</li> <li>amount Paid</li> <li>amountReserved</li> <li>amount Incurred</li> </ul> <p>It is required to fill out the following:</p> <ul style="list-style-type: none"> <li>effective date (if you don't have this, assume that it is 1 year prior to the quote request effective date)</li> <li>non Zero Claim Count</li> <li>lapse In Period (True or false); lapse Reason, if True</li> </ul> <p>Other fields are optional and will be used to provide add'l information for UW and Bind.</p>	<p>Put zero or the count &gt; 0 for each year that the customer had coverage; leave null for years without prior coverage.</p> <p>If the customer has no previous coverage, we expect all three of these fields to be <b>null</b>.</p> <ul style="list-style-type: none"> <li>claimCountYear1</li> <li>claimCountYear2</li> <li>claimCountYear3</li> </ul> <p><b>Ex</b> - If the client has 1 year of coverage with no claims, we expect a 0 for claimsCount Year 1 and nulls for years 2 and 3.</p>
<p>Other fields relevant to pricing</p>	<ul style="list-style-type: none"> <li>Payroll</li> <li>Class Codes</li> <li>Experience Modifier</li> <li>Blanket Waiver</li> </ul>	<ul style="list-style-type: none"> <li>Payroll</li> <li>Class Codes</li> <li>Experience Modifier</li> <li>Blanket Waiver</li> </ul>





# Critical data points for accurate pricing

## Quote Prior Carrier Examples

```
} "priorCarriers": [ Untitled-1 1 ●
1  "priorCarriers": [
2    {
3      "effectiveDate": "2019-08-19",
4      "expirationDate": "2020-08-19",
5      "totalPremium": 20420.75,
6      "lapseInPeriod": false,
7      "nonZeroClaimCount": 1,
8      "zeroClaimCount": 0,
9      "amountPaid": 10000,
10     "amountReserved": 10000,
11     "amountIncurred": 100,
12     "modificationFactor": 0.5,
13     "name": "Brownie Insurance",
14     "policyNumber": "1234567890"
15   },
16   {
17     "effectiveDate": "2021-08-19",
18     "expirationDate": "2022-08-19",
19     "totalPremium": 5000.75,
20     "lapseInPeriod": false,
21     "nonZeroClaimCount": 0,
22     "zeroClaimCount": 0,
23     "amountPaid": 0,
24     "amountReserved": 0,
25     "amountIncurred": 100,
26     "modificationFactor": 0.5,
27     "name": "Cookie Insurance",
28     "policyNumber": "0987654321"
```

This example would be considered 2 years of prior coverage. The more years of coverage provided, the more accurate of a price we can provide. Each block is considered a year of coverage.

## Price Indication Examples

```
{
  "state": "CA",
  "zip": "90210",
  "classCodes": [
    {
      "classCode": "5403",
      "payroll": 20000
    }
  ],
  "partnerAgentEmail": "partner@agent.com",
  "contactEmail": "contact@customer.com",
  "currentlyCovered": true,
  "claimCountYear1": 0,
  "claimCountYear2": 0,
  "claimCountYear3": 1,
  "effectiveDate": "2022-08-19",
  "emod": 1,
  "blanketWaiver": false
}
```

This Price Indication would show 3 years coverage with a loss claim in year 3. More years of coverage provided will give the most accurate pricing. If "claimCountYear#" is not shown

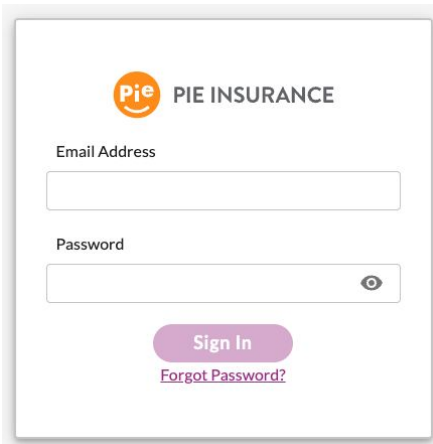


# Authentication

Current and new clients of Pie's external API can be set up with one or more API keys for obtaining actual credentials. The designated Admin for your account can create and revoke API keys or please [API@pieinsurance.com](mailto:API@pieinsurance.com) for help getting your first API Key and associated Agency code.

To access the API Keys for your account:

- Log into the portal as Admin.
  - The designated Partner Admin will be able to login with the credentials provided when your agency onboarded OR they may select "Forgot Password?" to receive an email with instructions on how to reset.



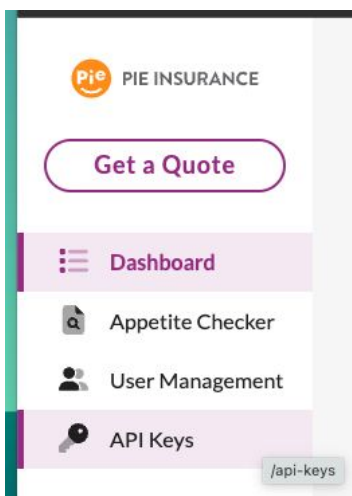
The image shows a sign-in form for Pie Insurance. At the top left is the Pie Insurance logo, consisting of an orange circle with 'Pie' inside and the text 'PIE INSURANCE' to its right. Below the logo are two input fields: 'Email Address' and 'Password'. The 'Password' field has a small eye icon to its right. Below the input fields is a purple 'Sign In' button and a link for 'Forgot Password?'.

Production Login: <https://partner.pieinsurance.com/sign-in>

Post-Prod Login: <https://partner.post-prod.pieinsurance.com/sign-in>

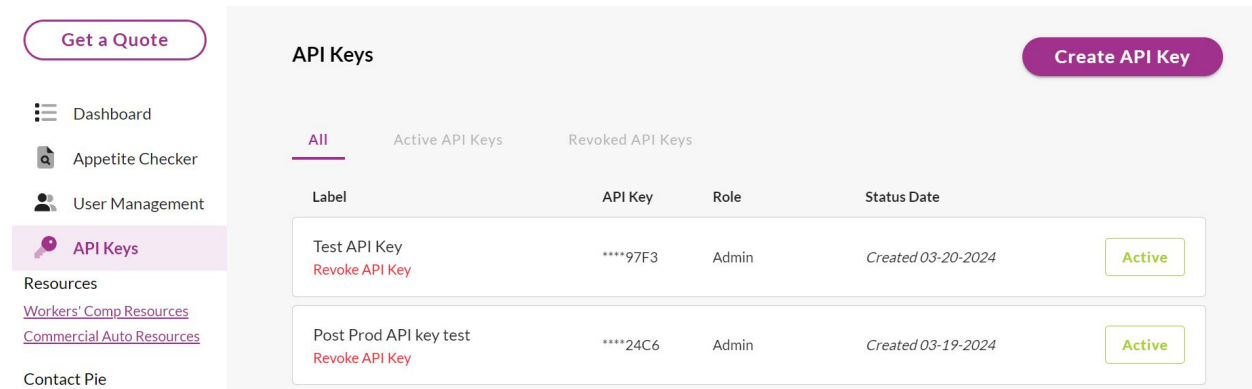
## View API Keys

- After successful login, select "API Keys" from the side menu



# Create and managing API Keys

## 1. Click Create API Key

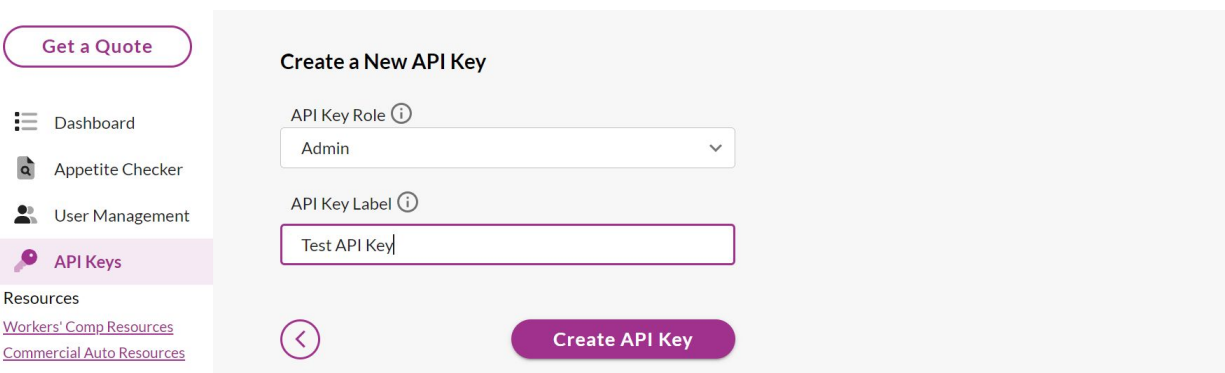


The screenshot shows the 'API Keys' management page. On the left is a navigation sidebar with a 'Get a Quote' button and menu items: Dashboard, Appetite Checker, User Management, API Keys (highlighted), Resources (with links for Workers' Comp Resources and Commercial Auto Resources), and Contact Pie. The main content area is titled 'API Keys' and has a 'Create API Key' button in the top right. Below the title are tabs for 'All', 'Active API Keys', and 'Revoked API Keys'. The 'All' tab is selected, displaying a table with the following data:

Label	API Key	Role	Status Date	
Test API Key <a href="#">Revoke API Key</a>	****97F3	Admin	Created 03-20-2024	<a href="#">Active</a>
Post Prod API key test <a href="#">Revoke API Key</a>	****24C6	Admin	Created 03-19-2024	<a href="#">Active</a>

## 2. For API key role select Admin

## 3. Name API Key Label

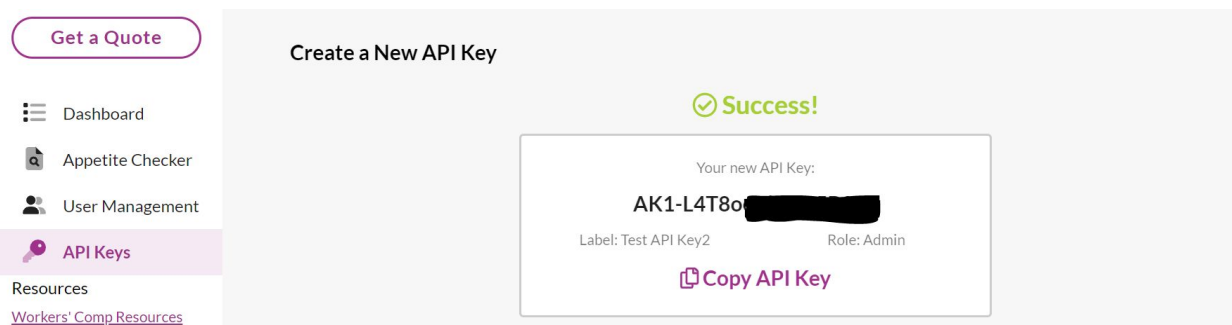


The screenshot shows the 'Create a New API Key' form. It includes a 'Get a Quote' button and the same navigation sidebar as the previous screenshot. The form has two main input fields: 'API Key Role' with a dropdown menu set to 'Admin', and 'API Key Label' with a text input field containing 'Test API Key'. At the bottom left is a back arrow icon, and at the bottom right is a 'Create API Key' button.

## 4. Click Create API Key

## 5. Click the checkbox to acknowledge that you'll be unable to view the full key again after leaving the page

## 6. You can now view your new API Key and copy it for implementation in your code



The screenshot shows the 'Create a New API Key' page with a success message. A green checkmark icon and the text 'Success!' are displayed. Below this, a white box contains the following information: 'Your new API Key:' followed by 'AK1-L4T8o' (where the last four characters are obscured by a black redaction bar). Below the key, it says 'Label: Test API Key2' and 'Role: Admin'. At the bottom of the box is a 'Copy API Key' button with a copy icon.

## 7. Click Back to API Keys

## 8. As Admin you can also revoke active keys by clicking Revoke API Key under the appropriate key.



# Responses

Conventional HTTP response codes are used to indicate the success or failure of an API request. In general, codes in the 2xx range indicate success. Codes in the 4xx range indicate an error that failed given the information provided (e.g., a required parameter was omitted, etc.). Codes in the 5xx range indicate a server error.

- 200** Successful request
- 201** Created (Quote endpoint)
- 202** Accepted (QuoteDocument endpoint)
- 400** Bad Request – Please refer to the “errors” field in the response
- 401** Unauthorized – Confirm your bearer token
- 415** Unsupported Media Type – Confirm your input data was valid and well-formed
- 500** Server Error – Intermittent issue (or a corner case we missed)
- 502** Bad Gateway - Intermittent issue, likely a code deployment

(This information is also documented on the [Pie Swagger page](#).)

*Our goal is to return clear validation messages for ease of development with our API. If you encounter any unexpected errors during your development process, please let us know so that we can continue to improve our API experience.*



# Detailed responses - quote endpoint

Valid requests to the **Quote** endpoint will generate a response with HTTP status code 201 for bindable or referred quotes, as indicated by the “bindStatus” field on the response. For bindable quotes, the response includes pricing, payment schedules, and a breakdown by class code. For referred quotes, the response indicates the refer reason.

**201** This value is the response to a well-formed request that Pie will potentially underwrite.

- **bindStatus : quotable:**
- **bindStatus : refer:** Indicates that Pie’s underwriting team will review the submission. As stated above, it is critical that supporting documentation (e.g., the ACORD 130) is supplied via the **Quote Document** endpoint.
- **referReasons:** In cases where **bindStatus is refer**, one or more reasons are included in the response. Refer reasons, at present, include:
  - Submitted payroll is below class code threshold.
  - Submitted loss history is below class code threshold.
  - Submitted governing class code requires review.
  - Estimated premium is above threshold.
  - Estimated schedule mod is above threshold.
  - Submitted experience modifier is too low.
  - Submitted loss history is below class code hazard group threshold.
  - Your partner account is set to refer.
  - General eligibility responses require review.
  - Industry eligibility responses require review.
  - No current coverage requires review.

Valid requests to the **Quote** endpoint will generate a response with HTTP status code 200 for declined quotes, as indicated by the “bindStatus” field on the response. Additional details can be found in the “declineReasons” field.

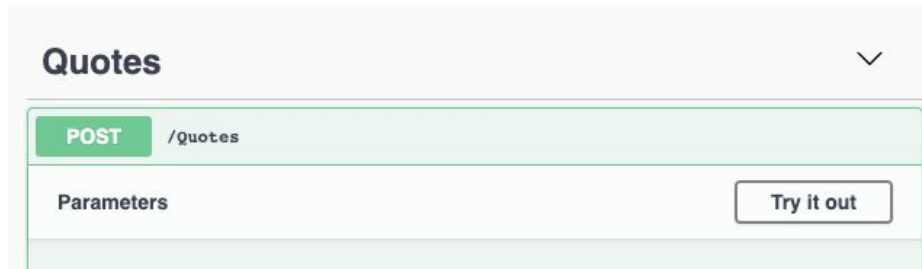
**200** This value is the response to a well-formed request that Pie cannot presently underwrite.

- **bindStatus : declined:** Indicates that the submission as received cannot be covered by Pie at present.
- **declineReasons:** In cases where **bindStatus is declined**, one or more reasons are included in the response.
  - Pie does not currently offer coverage in state {0}.
  - Pie does not currently offer coverage for class code {0}.
  - Pie does not currently offer coverage for class code {0}.
  - The experience mod provided exceeds the maximum amount that Pie is currently able to write.
  - Not enough years of prior continuous workers compensation coverage.
  - Payroll and loss history does not meet the minimum requirements for this class.
  - Policy premium exceeds the maximum amount that Pie is currently writing.
  - Pie is unable to sufficiently price this account to cover the exposures presented by this risk.
  - Does not meet general eligibility standards.
  - Does not meet industry eligibility standards.
  - The submitted total payroll is less than the minimum (\$1,000).
  - State in address must match address in parent submission object.



# Manual API testing with Swagger

Navigate to the [Pie Swagger page](#), click on **Post/Quotes** section, then click **Try it out**.



The image shows the Swagger UI for the 'Quotes' endpoint. At the top, the title 'Quotes' is displayed with a dropdown arrow. Below it, a green bar indicates the method 'POST' and the path '/quotes'. Underneath, there is a 'Parameters' section and a 'Try it out' button.

Enter the access\_token from the Postman response in the Bearer Authorization box with the word Bearer in front of it (see example below).

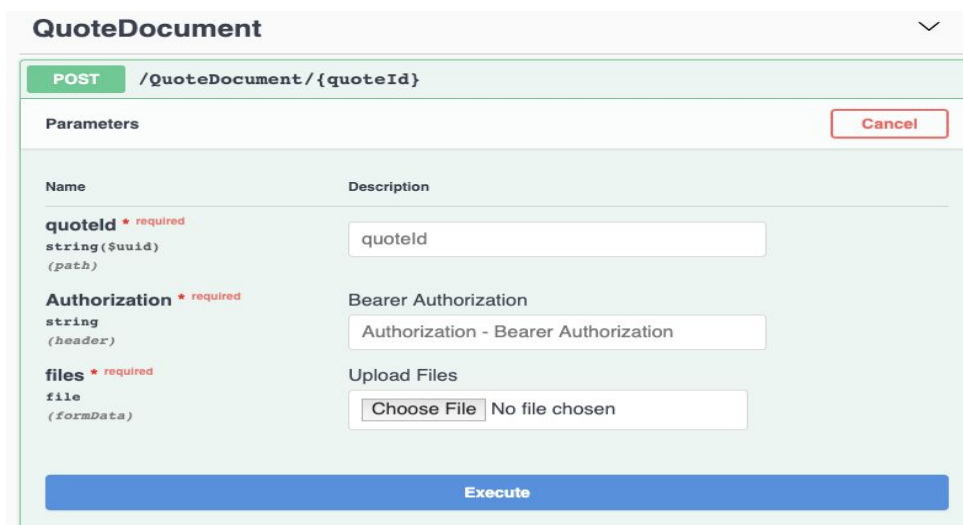


The image shows the 'Authorization' configuration in Swagger UI. It is labeled 'Authorization \* required' and 'string (header)'. The 'Bearer Authorization' section contains a text box with the value 'Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpX'. At the bottom, there is a large blue 'Execute' button.

You can now click **Execute** to submit the sample request and obtain a response from the Quote API. From here, you can edit the sample or write/paste in your test query.

Swagger details the models used for input, the acceptable input options, and the universe of response errors. This detail will be useful in troubleshooting with state-specific settings and options.

After generating a test response, use the ID from the response to test the QuoteDocument endpoint. Click the **QuoteDocument** section on Swagger, then **Try it out**. Enter the quoteId and Bearer Token, then upload a document. A successful request will generate a response consisting of a 202 and a quote URL.



The image shows the Swagger UI for the 'QuoteDocument' endpoint. The title is 'QuoteDocument'. The method is 'POST' and the path is '/QuoteDocument/{quoteId}'. There is a 'Parameters' section with a 'Cancel' button. The parameters are listed in a table:

Name	Description
<b>quoteId</b> * required string (\$uuid) (path)	quoteId
<b>Authorization</b> * required string (header)	Bearer Authorization Authorization - Bearer Authorization
<b>files</b> * required file (formData)	Upload Files Choose File No file chosen

At the bottom, there is a large blue 'Execute' button.



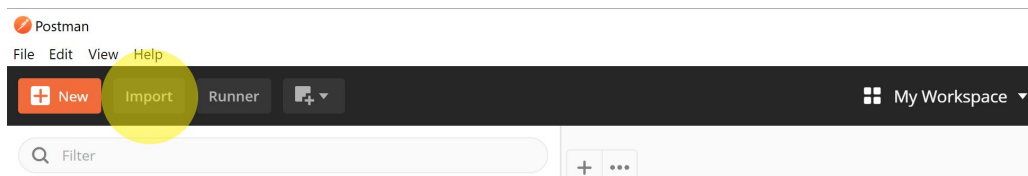
# Developing your API using Swagger

The [Swagger](#) site has a built-in example for a full quote request. This example, plus the models on Swagger, is a good starting point to develop against our API.

The best way to get the full example is to copy it from [Swagger](#) under the Quotes section. If you would like more detailed directions for downloading a full example, follow the Postman [Swagger Import Walkthrough](#) directions.

Using [Postman](#), you can import the swagger.json into a collection with the sample request. The following images and instructions will show you a walkthrough of importing the [swagger.json](#). The yellow circles on the image point out the buttons and input fields that the user will use to get through the process.

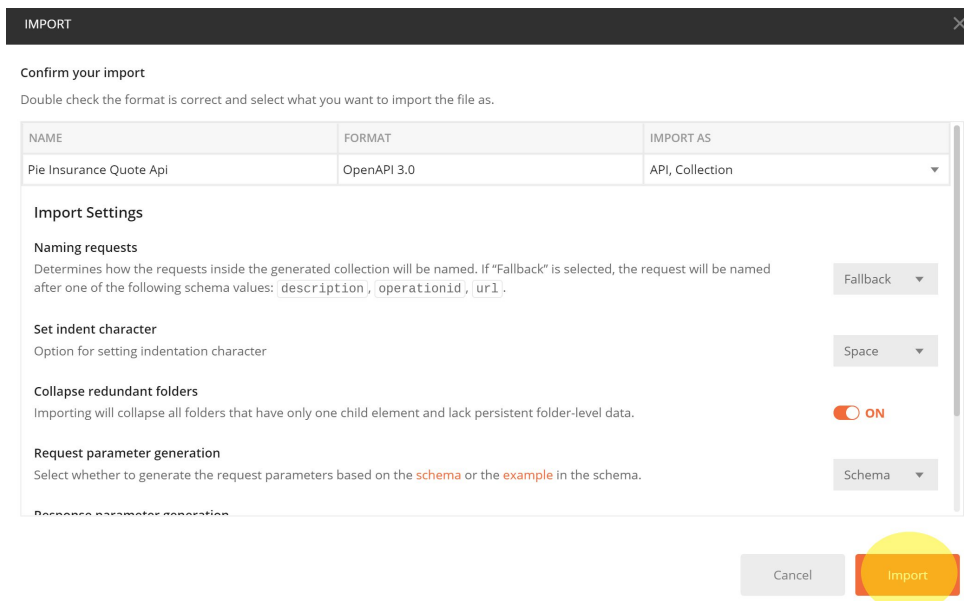
You begin the import process by clicking the Import button.



An import dialog should appear. Click the link tab and input the [swagger.json](#) url. Click Continue to proceed.



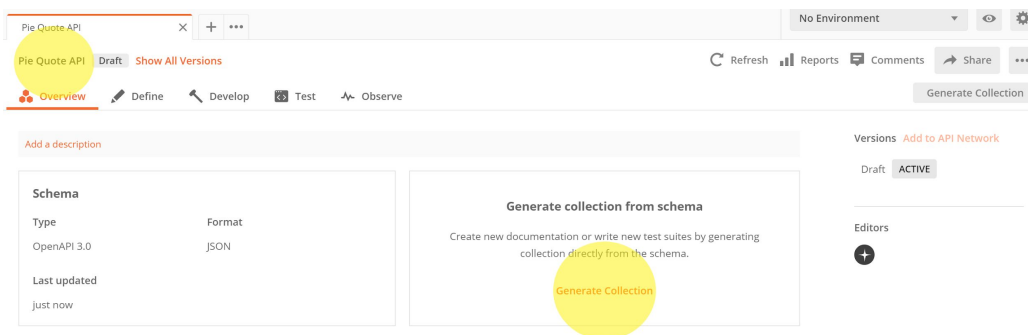
A new dialog will appear. Accept the default configuration and click Import.



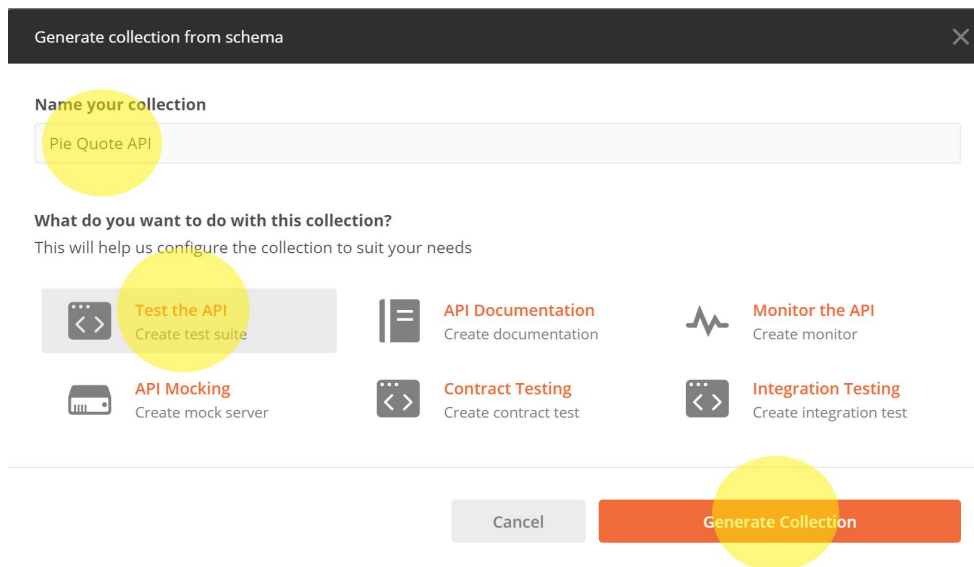


# Developing your API using Swagger, cont'd

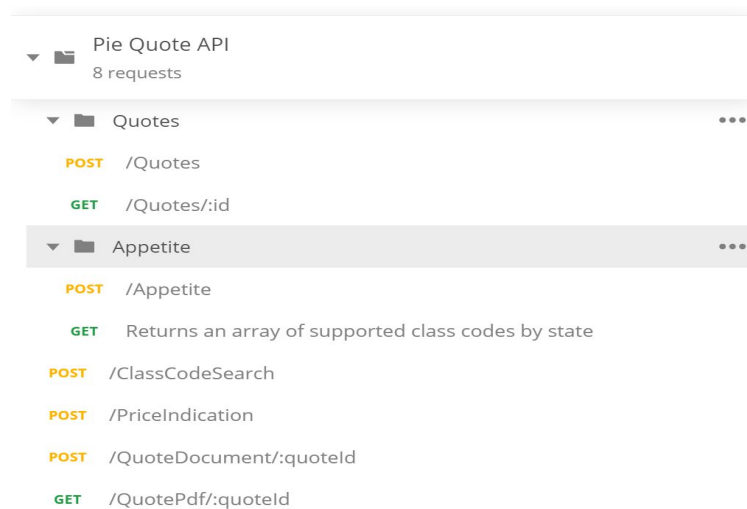
An API configuration should appear. You will have an option to name the API. Once you have finished naming the API, click the Generate Collection link.



A Collection configuration dialog will appear. You will be able to name the Collection. Select the Test the API option and click Generate Collection.



A collection should appear on the left panel with the name you specified in the previous dialog. The collection will contain the endpoints and tests defined in the [swagger.json](#).



# FAQ's

## Deductibles

**Our API does not currently support deductibles.** Pie underwriters may offer a deductible later in the process based on an applicant's ability to pay, and of course not to exceed the annual premium.

## Policy Effective Dates and Backdating

Our API requires policy effective dates to be in the future (next day at the earliest). Backdating requires subsequent underwriting review and approval.

## Waivers

**Specific waivers are not currently supported by our API.** Your agents can discuss waivers on a case-by-case basis with our underwriters prior to issuance.

## Technical Limitations

Our API does not currently collect information about **locations across multiple U.S. states**. Requests that cross multiple states can be made directly in our Partner Portal.

Our API collects data about **owners and officers**, but the price returned does not reflect inclusion / exclusion for them. That information will be handled by our underwriting team at the time of bind.

The API supplies a new quote for each successful request. Once the Quote API generates a quote, there is currently *no way to modify* that quote without sending a request for an update to the underwriting team. The QuoteDocument endpoint enables the uploading of supporting documentation (e.g., ACORD 130, loss runs, or supplemental applications), which appends documents to a quote for Pie's underwriting team to review.

## Deep Link to Quote

Once a Quote has been obtained, the agent will need to bridge into the portal to finish the bind or to make any additional edits to the quote. To do so this deeplink can be used in your UI to bridge.

<https://partner.{environment}.pieinsurance.com/agency-flow/decision/{decisionID}>

The Decision ID is returned at the end of the full quote in the response. This only works for QUOTE, not Price Indication

```
    ],  
    "id": "08da6a88-90f2-647f-698b-b500011ee65f",  
    "bindStatus": "Quotable"
```





Thank You!