Embedded API

API Integration Guide

v1.1.0

29 September 2026

QUOTALL

Contents

INTRODUCTION 4

DESCRIPTION 4

TESTING 4

VERSIONING 5

GENERAL PRINCIPLES 5

PROTOCOL 5
AUTHENTICATION 5

DESCRIPTION OF INSURANCE PROCESS 6

NEW BUSINESS 6
RENEWALS 6

BRIEF DESCRIPTION OF ENDPOINTS 8

CONCEPT AND PROCESSING SEQUENCE 8
SERVICE AVAILABILITY 9
AUTHORISE 9
POLICIES 10
DOCUMENTS 11
CUSTOMERS 12
CLAIMS 13
SUBSCRIPTIONS 14

SERVICE AVAILABILITY ENDPOINT 15

DETERMINE WHETHER THE SERVICE IS AVAILABLE 15

AUTHENTICATION ENDPOINTS 16

OBTAIN AN AUTHENTICATION TOKEN 16
HOW TO USE THE AUTHENTICATION TOKEN 18

POLICY ENDPOINTS 19

POLICY INFORMATION 19
CREATE A NEW POLICY 21
CANCEL AN EXISTING POLICY 24
CONFIRM PAYMENT HAS BEEN TAKEN FOR A POLICY RENEWAL 25
SET AUTO-RENEW STATUS FOR AN EXISTING POLICY 27

DOCUMENT ENDPOINTS 28

RETURN A POLICY WORDINGS DOCUMENT 28
RETURN AN INSURANCE PRODUCT INFORMATION DOCUMENT (IPID) 29
RETURN A POLICY SCHEDULE DOCUMENT 30
UPLOAD A CLAIMS DOCUMENT 31

CUSTOMERS ENDPOINTS 32

RETURN ALL POLICIES FOR A CUSTOMER 32
RETURN A POLICY FOR A CUSTOMER 34
UPDATE CUSTOMER INFORMATION 36

CLAIMS ENDPOINTS 38

RETURN THE QUESTIONS REQUIRED TO CREATE A CLAIM 38
CLAIM QUESTION PROPERTIES 41
CLAIM QUESTION ANSWER TYPES 42
REGISTER A CLAIM FOR AN EXISTING POLICY 42
RETURN THE STATUS OF CLAIMS ON A POLICY 45
RETURN THE URL OF THE HOSTED CLAIMS PORTAL 47

DATA DICTIONARY 48

APPENDIX 53

POSTMAN COLLECTION 53

AMENDMENTS 54

Introduction

This document is a guide to integration with the Quotall Embedded Insurance API.

Once integrated it will be possible to create new business policies cancel policies renew policies and record claims against policies using the API endpoints provided.

Description

The Quotall Embedded Insurance API is a REST API.

Briefly REST API's work with JSON notation following these steps:

1. Your system sends an HTTP request to our server specifying the URL of the endpoint to call and the method (POST). The request will also include data in JSON format in the body of the call.

2. Our server receives the request and processes it then sends an HTTP response back to your system containing a status code that indicates whether the request was successful or not and where appropriate a body in JSON containing relevant data.

Testing

We recommend testing API calls in Postman.

The raw JSON of a pre-configured Postman "Collection" which can be directly imported into Postman can be provided on request.

Endpoint calls can be made directly from Postman to create cancel and record claims against policies.

Versioning

The Quotall API's are versioned. The current version is: v1.0

Please Note:

- These version numbers form part of the endpoint URL.
- Versions are subject to change; warning will be given should we require you to use an updated version.

General Principles

Protocol

All requests to the API must be made using the HTTPS protocol.

Authentication

The API uses JWT Tokens to authenticate API calls.

Prior to calling the endpoints a call to an Authentication endpoint is required to obtain a token.

This token should then be supplied in the header of requests to the other endpoints.

The lifetime of a token is 12 hours. Once a token has expired a fresh one will need to be obtained. **Please note**: the token lifetime is subject to change; warning will be given should it change.

To call the authentication endpoint the following identifying data supplied by Quotall is required:

- Client Id
- Client Secret

These may differ depending on whether calls are made to the staging system (for testing) or the production system.

Description of insurance process

New Business

1. Call **policies/policyInformation** on display of the product detail page and display the returned price in your insurance section. This is the price that should be charged if the customer buys the insurance.
2. After successful payment for the goods and insurance call **policies/create** to instantiate the insurance policy.

Renewals

By default, all insurance policies automatically renew on the day before the expiry date returned by the **policies/create** endpoint.

There are two conditions in which a policy will not automatically renew:

1. Payment for the renewal has not been taken by the renewal date.
2. The customer has indicated that they do not require the automatic renewal.

In respect of (1) your system must take payment for the renewal by the day before the renewal takes place, that is two days prior to the policy expiry date.

Timeline

Five days prior to the expiry of the policy, you must:

a) Check the auto-renew status of the policy, if "off", do nothing, otherwise:
b) Call the **policies/policyInformation** Api endpoint to get the renewal premium. This step must be done as the premium could differ from the premium of the renewing policy. The renewingPolicyReference field must be populated for renewal price requests. It will then only give a price if the policy is renewable. This prevents payment being taken for policies which cannot be renewed, for example, if they've been cancelled.
c) Take payment using the stored payment **details/token**.
d) Call the **policies/renew** Api endpoint to send us the payment reference which serves to inform us that payment has been taken for the renewal as well as providing an audit trail. The Api checks that a renewal quote exists for this policy, if not returns a failure message. Note: If you do this too soon (before the five days) the renewal quote may not yet exist.
e) Our Api stores the payment reference you send on the renewal quote which informs us that the payment has been taken and the renewal can go ahead.

f) The **policies/renew** endpoint returns the new policy reference, calculated by the Api from the old one (even though it doesn't strictly exist yet) and the expiry date of the renewed policy.

If you have not sent us the payment reference at least two days prior to the expiry date the policy will not renew.

In respect of (2) regulatory requirements obligate you to provide a mechanism by which your customers can switch off automatic renewals. We suggest displaying an on/off toggle alongside the displayed insurance policy details on the customer's account page.

When the customer toggles the switch call the **policies/autoRenew** endpoint to indicate whether the customer has either switched off or switched on the auto-renew facility.

If its status is off on the day before the expiry date the policy will not renew.

Brief description of endpoints

Concept and processing sequence

auth/identity should be called initially to retrieve a JWT token used to authenticate all subsequent endpoint calls.

The token has a lifetime of 12 hours and can be re-used until it's expiry date/time.

policies/policyInformation is called initially to get a "quotation" detailing the insurance premium or price together with the name and a short summary of the insurance policy.

It is expected that this information be surfaced to the customer on the checkout page.

The optional *renewingPolicyReference* property should only be populated if this request is in respect of a renewal price.

policies/create is called if the customer has indicated on your checkout page that they wish to purchase the policy and they have successfully paid for both the product and the insurance.

On success our system will send the customer a policy purchase confirmation email branded for your organisation. This email has PDF copies of the Policy Schedule (insurance certificate) and the Policy Wordings (describes the benefits and terms and conditions of the policy) attached.

policies/renew is called to confirm that payment has been successfully taken for a forthcoming automatic renewal. **Note**: Returns a new policy reference replacing the previous one. This new policy reference should be used in all future endpoint calls which require it relating to the policy.

policies/autoRenew is called to switch on or off automatic renewals. By default an insurance policy will be automatically renewed the day before it's expiry date – if payment has been taken.

Other endpoints are provided to maintain and administer the created insurance policy or policies if a customer has purchased multiple products each with their own insurance policy.

Details of the Policy or Policies can be surfaced on your customers account page and you can provide functionality to obtain documents make a claim and renew or cancel the policy using these additional endpoints.

Service Availability

GET embapi

Returns true or false depending on the availability of the service.

Authorise

GET auth/identity

Returns a JWT Token used to authenticate all other endpoints.

Policies

POST policies/policyinformation

Get the price and other information about the insurance product to display to the customer.

POST policies/create

Create an insurance policy.

POST policies/cancel/{policyReference}

Cancel an existing insurance policy.

POST policies/renew

Confirm payment has been taken for an automatic renewal. Returns the new policy reference and expiry date of the renewed policy.

POST policies/autorenew/{policyReference}/on|off

Switch automatic renewals on or off.

Documents

GET documents/policyDocument/{policyTypeId}

Returns a document describing the benefits and terms and conditions of the policy. This document is not personalised. It can be surfaced prior to purchase. The pdf document is returned as an array of bytes.

This can be saved as a file or returned directly to the customer's browser.

GET documents/insuranceProductInformationDocument/{policyTypeId}

Returns an Insurance Product Information Document (IPID) describing the benefits and terms and conditions of the policy. Often included in the Policy Wordings Document but may be provided separately. This document is not personalised. It can be surfaced prior to purchase. The PDF document is returned as an array of bytes.

This can be saved as a file or returned directly to the customer's browser.

GET documents/policySchedule/{policyReference}

Returns the "Policy Schedule" document after the policy has been purchased.

This is in effect the certificate of insurance. The pdf document is returned as an array of bytes.

This can be saved as a file or returned directly to the customer's browser.

POST documents/claimsFileUpload/{claimReference}

Accepts a file uploaded by a customer in the event of a claim. This might be an invoice for the goods a police report or similar in support of the claim.

Customers

GET customers/policies/{email}

Returns policy details including the expiry date, auto-renew status and premium information of all policies held by a customer.

GET customers/policy/{policyReference}

Returns policy details including the expiry date, auto-renew status and premium information of a specific policy held by a customer.

PUT customers/updateCustomer/{customerId}

Updates customer information such as change of email address.

Claims

GET claims/questions/{policyReference}

When a customer makes a claim on their insurance policy the insurer requires specific information from the customer in order to process their claim.

This endpoint returns information about the questions that the customer should be asked.

POST claims/create

Creates a claim on the customers policy.

GET claims/status/{policyReference}

Returns details, including the status, of all claims on a Policy.

GET claims/portal/{policyReference}

Returns the URL of a hosted claims creation page. This page will be branded for your organisation but hosted on our servers.

This is an alternative to building your own Claims “First notification of loss” form using information returned the in claims/questions endpoint.

Subscriptions

A selection of endpoints supporting various cloud services providing the means to subscribe to events which provide real-time updates on claims and renewals.

These will be made available sometime during or after October 2025.

Service Availability Endpoint

Determine whether the service is available

REQUEST

Method: GET

Staging URL: https://embapistaging.quotall.com/v1.0/embapi

Production URL: https://embapi.quotall.com/v1.0/embapi

Headers: None

RESPONSE

HTTP 200 OK – denotes service is available. HTTP 503 ServiceUnavailable – service is not available.

Authentication Endpoints

Obtain an authentication token

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/auth/identity

Production URL: https://embapi.quotall.com/v1.0/auth/identity

Headers:

Name	Value
Content-Type	"application/json"
client_id	*Quotall to allocate*
client_secret	*Quotall to allocate*

RESPONSE

Success example


```
{
   token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6NTh9.eyJyb2xlIjoiMTgiLCJ1bmlxdWVfbmFtZSI6IlJlbnRNeSIsIm5iZiI6MTY5MDk4MTI1MSwiZXhwIjoxNjkxMDI0NDUxLCJpYXQiOjE2OTA5ODEyNTEsImlzcyI6Imh0dHBzOi8vYXBpLnF1b3RhbGx1YXQuY28udWsiLCJhdWQiOiJyZW50bXktYXBpLnF1b3RhbGwuY29tIiwicHJvamVjdElkIjp1OCwiY29tcGFueUNvZGUiOiJSRU5UTSIsInByb2plY3RDb21wYW55Q29kZSI6IlJFTlRNIiwiaW50ZXJmYWNlQ17kZSI6IkJJIiwiYWNjb3VudCI6IjY0Nzg4IiwidXNlcklkIjoicmVudG15YnJva2VyQHF1b3RhbGwuY29tIiwidXNlclR5cGUiOiJCUk9LIiwiZm9yZW5hbWUiOiJSZW50TXkiLCJzdXJuYW1lIjoiQnJva2VyIn0.qWmThl-Il7aoFWRsl1oC1V1kt67e3s-Z762x43t0I3o"
}
```


Invalid example


```
{
   code: "400_137",
   text: "Incorrect client_id or client_secret",
   description: "The client_id, client_secret, or both, provided in a header of the call are invalid. Please check the values.",
   technical: "None available",
   helpPageUrl: "https://Quotall.co.uk/api"
}
```

How to use the authentication token

The token should be added to the authorization header of each subsequent endpoint called:

Name: Authorization

Content: "Bearer *token*"

Example content:

"Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6NTh9.eyJyb2xlIjoiMTgiLCJ1bmlxdWVfbmFtZSI6IlJlbnRNeSIsIm5iZiI6MTY5MDk4MTI1MSwiZXhwIjoxNjkxMDI0NDUxLCJpYXQiOjE2OTA5ODEyNTEsImlzcyI6Imh0dHBzOi8vYXBpLnF1b3RhbGx1YXQuY28udWsiLCJhdWQiOiJyZW50bXktYXBpLnF1b3RhbGwuY29tIiwicHJvamVjdElkIjp1OCwiY29tcGFueUNvZGUiOiJSRU5UTSIsInByb2plY3RDb21wYW55Q29kZSI6IlJFTlRNIiwiaW50ZXJmYWNlQ17kZSI6IkJJIiwiYWNjb3VudCI6IjY0Nzg4IiwidXNlcklkIjoicmVudG15YnJva2VyQHF1b3RhbGwuY29tIiwidXNlclR5cGUiOiJCUk9LIiwiZm9yZW5hbWUiOiJSZW50TXkiLCJzdXJuYW1lIjoiQnJva2VyIn0.qWmThl-Il7aoFWRsI1oC1V1kt67e3s-Z762x43t0I3o"

Policy Endpoints

Policy Information

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/policies/policyInformation

Production URL: https://embapi.quotall.com/v1.0/policies/policyInformation

Headers:

Name	Value
Content-Type	"application/json"
Authorization	"Bearer *token*"

Body: (application/json) with example data

```
{
  productCategory: "WATCHES",
  productSubCategory: "Metal Straps",
  productRetailPrice: 6999.99,
  productSalePrice: 4490.50,
  renewingPolicyReference: "0001CAUL23X1"
}
```

Note: renewingPolicyReference should only be populated if the request is for a renewal price.

RESPONSE

Success example


```
{
   policyTypeId: "113",
   policyName: "Article Insurance",
   policyInformation: {
      policySummary: "Protect your watch against theft, loss and damage",
      annualPrice: 235.20,
      iptAmount: 25.20,
      monthlyPrice: 19.60,
      currency: "GBP"
   }
}
```


Invalid example


```
{
   code: "400",
   text: "Invalid input field",
   description: "Product Category does not have a valid value",
   technical: "None available",
   helpPageUrl: "https://Quotall.co.uk/api"
}
```

Create a new Policy

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/policies/create

Production URL: https://embapi.quotall.com/v1.0/policies/create

Headers:

Name	Value
Content-Type	"application/json"
Authorization	"Bearer *token*"

Body: (application/json) with example data

```
{
  productCategory: "WATCHES",
  productSubCategory: "Metal Straps",
  productRetailPrice: 6999.99,
  productSalePrice: 4490.50,
  productName: "Rolex Oyster",
  productId: "20105585",
  orderReference: "ORD123456",
  orderItemId: "5625594",
  customerDetails: {
    email: "mail@mail.com",
    customerId: "33489031",
    firstName: "Robert",
    lastName: "Smith",
    address: {
      addressLine1: "1 The Avenue",
      addressLine2: "Highbury",
      townCity: "London",
      postCode: "N6 5PT"
    }
  }
}
```

RESPONSE

Success example


```
{
   policyReference: " 0001GEYB24X0 ",
   productName: "Rolex Oyster",
   productId:  "20105585",
   orderReference: "ORD123456",
   orderItemId: "5625594",
   policyExpiryDate:  "2025-07-19",
}
```


Invalid example


```
{
   code: "400",
   text: "Invalid input field",
   description: "Product Category does not have a valid value",
   technical: "None available",
   helpPageUrl: "https://Quotall.co.uk/api"
}
```

Cancel an existing Policy

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/policies/cancel/{policyReference}

Production URL: https://embapi.quotall.com/v1.0/policies/cancel/{policyReference}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example


```
{
    refundAmount: 302.40,
    currency: "GBP"
}
```


Invalid – example


```
{
    code: "400",
    text: "Invalid input field",
    description: "A policy with that reference does not exist",
    technical: "",
    helpPageUrl: ""
}
```

Confirm payment has been taken for a Policy Renewal

REQUEST

Method: POST

Staging URL:

https://embapistaging.quotall.com/v1.0/policies/renew

Production URL: https://embapi.quotall.com/v1.0/policies/renew

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

Body: (application/json) with example data

```
{
  policyReference: "0001GEYB24X0",
  paymentReference: "ABC 123456 XCS"
}
```

RESPONSE

Success – example

```
{
  policyReference: "0001GEYB24X1",
  policyExpiryDate:  "2026-07-19"
}
```

Invalid – example


```
{
   code: "400",
   text: "Invalid input field",
   description: "Policy with that reference does not exist",
   technical: "",
   helpPageUrl: ""
}
```

Set Auto-Renew Status for an existing Policy

REQUEST

Method: POST

Staging URL:

https://embapistaging.quotall.com/v1.0/policies/autorenew/{policyReference}/on|off

Production URL:
https://embapi.quotall.com/v1.0/policies/autorenew/{policyReference}/on|off

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example

```
200 OK
```

Invalid – example

```
{
   code: "400",
   text: "Invalid parameter",
   description: "Status must have a value of either 'on' or 'off'",
   technical: "",
   helpPageUrl: ""
}
```

Document Endpoints

Return a Policy Wordings Document

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/documents/policyDocument/{policyTypeId}

Production URL: https://embapi.quotall.com/v1.0/documents/policyDocument/{policyTypeId}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success

The PDF policy wordings document in bytes, intended to be directed to the user's browser.

Invalid – example

```
{
   code: "400",
   text: " Bad Request",
   description: " policyTypeId does not exist",
   technical: "",
   helpPageUrl: ""
}
```

Return an Insurance Product Information Document (IPID)

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/ documents/ documents/insuranceProductInformationDocument/{policyTypeId}

Production URL: https://embapi.quotall.com/v1.0/ documents/insuranceProductInformationDocument/{policyTypeId}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success

The PDF policy IPID document in bytes, intended to be directed to the user's browser.

Invalid – example

```
{
   code: "400",
   text: " Bad Request",
   description: " policyTypeId does not exist",
   technical: "",
   helpPageUrl: ""
}
```

Return a Policy Schedule Document

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/ documents/ policySchedule/{policyReference}

Production URL: https://embapi.quotall.com/v1.0/ documents/ policySchedule/{policyReference}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success

The PDF policy schedule document in bytes, intended to be directed to the user's browser.

Invalid – example


```
{
   code: "400",
   text: " Bad Request",
   description: " Unable to fetch policy schedule document for 0001FYOC24X0",
   technical: "",
   helpPageUrl: ""
}
```

Upload a Claims Document

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/ documents/claimsFileUpload/{claimsReference}

Production URL: https://embapi.quotall.com/v1.0/ documents/claimsFileUpload/{claimsReference}

Headers:

Name	Value
"Content-Type"	"multipart/form-data"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success

An Html Status Code 200 response

Invalid – example


```
{
   code: "400",
   text: " Bad Request",
   description: "Unable to upload claim file for claim reference 1568",
   technical: "",
   helpPageUrl: ""
}
```

Customers Endpoints

Return all policies for a customer

REQUEST

Method: GET

Staging URL: https://embapistaging.quotall.com/v1.0/customers/policies/{email}

Production URL: https://embapi.quotall.com/v1.0/customers/policies/{email}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example


```
[
  {
    policyReference: "0001FYRM24X0",
    productName: "Rolex Oyster",
    productId: "20105585",
    orderItemId: "5625594",
    policyExpiryDate: "2025-08-15",
    policyStatus: "active",
    grossPremium: 235.20,
    iptAmount: 25.20,
    autoRenew: "on"
  },
  {
    policyReference: "0001FYRN24X0",
```

```
    productName: "Rolex Submariner",
    productId: "20105585",
    orderItemId: "5625595",
    policyExpiryDate: "2025-08-15",
    policyStatus: “expired”,
    grossPremium: 120.63,
    iptAmount: 12.92,
    autoRenew: “off”
  },
  {
    policyReference: "0001FYRN24X0",
    productName: "Rolex DatePlus",
    productId: "20105592",
    orderItemId: "5625568",
    policyExpiryDate: "2025-10-20",
    policyStatus: “cancelled” ”,
    grossPremium: 80.63,
    iptAmount: 8.64,
    autoRenew: “on”
  }
]
```


Invalid – example


```
{
  code: "204",
  text: "No Content",
  description: "No policies found for mail@mail.com",
  technical: "",
  helpPageUrl: ""
}
```

Return a policy for a customer

REQUEST

Method: GET

Staging URL:
https://embapistaging.quotall.com/v1.0/customers/policy/{policyReference}

Production URL: https://embapi.quotall.com/v1.0/customers/policy/{policyReference}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example


```
{
  policyReference: "0001FYRM24X0",
  productName: "Rolex Oyster",
  productId: "20105585",
  orderItemId: "5625594",
  policyExpiryDate: "2025-08-15",
  policyStatus: "active" ",
  grossPremium: 235.20,
  iptAmount: 25.20,
  autoRenew: "on"
}
```


Invalid – example


```
{
  code: "400",
```

```
    text: " Invalid input field",
    description: " A policy with that reference does not exist ",
    technical: "",
    helpPageUrl: ""
}
```

Update customer information

REQUEST

Method: PUT

Staging URL:
https://embapistaging.quotall.com/v1.0/customers/updateCustomer/{customerId}

Production URL:
https://embapi.quotall.com/v1.0/customers/updateCustomer/{customerId}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: (application/json) with example data

```
{
   customerDetails: {
     email: "mail@mail.com",
     customerId: "33489031",
     firstName: "Robert",
     lastName: "Smith",
     address: {
       addressLine1: "1 The Avenue",
       addressLine2: "Highbury",
       townCity: "London",
       postCode: "N6 5PT"
    }
  }
}
```

RESPONSE

Success – example


```
{
    email: "mail@mail.com",
    customerId: "33489031",
    firstName: "Robert",
    lastName: "Smith",
    address: {
      addressLine1: "1 The Avenue",
      addressLine2: "Highbury",
      townCity: "London",
      postCode: "N6 5PT"
  }
}
```


Invalid – example


```
{
  code: "400",
  text: " Invalid input field",
  description: " A customer with that Id does not exist ",
  technical: "",
  helpPageUrl: ""
}
```

Claims Endpoints

Return the questions required to create a claim

REQUEST

Method: GET

Staging URL:
https://embapistaging.quotall.com/v1.0/claims/questions/{policyReference}

Production URL: https://embapi.quotall.com/v1.0/claims/questions/{policyReference}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example


```
[
  {
    "questionId": 23,
    "questionText": "Claim Type",
    "isMandatory": true,
    "hideOnLoad": false,
    "regExValidation": "",
    "validationErrorMessage": "",
    "answerType": "fixedOptions",
    "options": [
      {
        "code": "DAM",
        "text": "Damaged (repairable)",
```

```
            "showHideQuestions": []
        },
        {
            "code": "DEST",
            "text": "Destroyed (irreparable)",
            "showHideQuestions": []
        },
        {
            "code": "LOST",
            "text": "Lost (not stolen)",
            "showHideQuestions": []
        },
        {
            "code": "STOL",
            "text": "Stolen",
            "showHideQuestions": []
        }
    ],
    "answerOptions": []
},
{
    "questionId": 40,
    "questionText": "Date of Loss",
    "isMandatory": true,
    "hideOnLoad": false,
    "regExValidation": "",
    "validationErrorMessage": "",
    "answerType": "date",
    "options": [],
    "answerOptions": []
},
{
    "questionId": 41,
    "questionText": "Description of circumstances of loss",
    "isMandatory": true,
    "hideOnLoad": false,
    "regExValidation": "",
    "validationErrorMessage": "",
    "answerType": "multiLineText",
    "options": [],
    "answerOptions": []
```

```
  },
  {
    "questionId": 46,
    "questionText": "Have you reported this to the police?",
    "isMandatory": true,
    "hideOnLoad": false,
    "regExValidation": "",
    "validationErrorMessage": "",
    "answerType": "options",
    "options": [
      {
        "code": "17",
        "text": "No",
        "showHideQuestions": []
      },
      {
        "code": "16",
        "text": "Yes",
        "showHideQuestions": []
      }
    ],
    "answerOptions": []
  }
]
```


Invalid – example


```
{
  code: "400",
  text: "Invalid input field",
  description: "Policy with that reference does not exist",
  technical: "",
  helpPageUrl: ""
}
```

Claim Question properties

The properties provide information on the display and validation of the questions. Not all questions have all properties, any that are empty for a question can be safely ignored.

Value	Description
questionId	The Id of the question to identify the question in the claims/create endpoint.
questionText	The text to display alongside the question.
isMandatory	Whether an answer to this question must be provided.
hideOnLoad	Whether this question should be initially hidden before being revealed when another question is answered with a certain value.
regExValidation	A regular expression to validate that the given answer is in the correct format.
validationErrorMessage	A suitable message to display if the given answer fails validation.
answerType	The type of answer required, see table below.
options or fixedOptions	A list of possible answers to this question.
showHideQuestions	The other questions to show or hide when this question is answered with a specific option.
answerOptions	The options for answers to this question which are derived from an answer in the policy quote.

Claim Question answer types

The answer types indicate the type of data that is expected as an answer to the question. These are the possible values:

Value	Description	Suggested Html tag
singleLineText	A short text answer	<input>
singleLineTextCapitalised	A short text answer with each word capitalised. The endpoint will capitalise answers in the response for you.	<input>
singleLineTextUppercase	A short text answer with each letter capitalised. The endpoint will capitalise answers in the response for you.	
multiLineText	A long text answer perhaps containing line breaks.	<textarea>
decimalNumber	A decimal number usually a monetary amount.	<input>
wholeNumber	An integer	<input>
date	A date in the format yyyy-mm-ddThh:mm:ssZ (UTC ISO 8601)	<input>
options or fixedOptions	Answer must be of fixed type from the list provided.	<select>
title	This is simply text which you can optionally use as sub-heading on the claim form.	<h4>

Register a Claim for an existing Policy

REQUEST

Method: POST

Staging URL: https://embapistaging.quotall.com/v1.0/claims/create

Production URL: https://embapi.quotall.com/v1.0/claims/create

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: (application/json) with example data

```
{
  policyReference: "0001FMLF24T0",
  questionAnswers: [
    {
      questionId: 23,
      answer: "LOST"
    },
    {
      questionId: 40,
      answer: "2022-08-06T00:00:00Z"
    },
    {
      questionId: 41,
      answer: "Can't find it anywhere"
    },
    {
      questionId: 46,
      answer: "17"
    }
  ]
}
```

RESPONSE

Success – example

```
{
```

```
  policyReference: "0001FYRO24X0",
  claimReference: 1007,
  claimStatus: "Claim Received",
  claimDate: "2024-08-16",
  statusLastUpdated: "2024-08-16"
}
```


Invalid – example


```
{
  code: "400",
  text: "Invalid input field",
  description: "Policy with that reference does not exist",
  technical: "",
  helpPageUrl: ""
}
```

Return the status of claims on a policy

REQUEST

Method: GET

Staging URL: https://embapistaging.quotall.com/v1.0/claims/status/{policyReference}

Production URL: https://embapi.quotall.com/v1.0/claims/status/{policyReference}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example


```
[
        {
          policyReference: "0001GKDG24X0",
          claimReference: 1211,
          claimStatus: "Awaiting Information",
          claimDate: "2024-11-25",
          statusLastUpdated: "2024-11-30"
        },
        {
          policyReference: "0001GKDG24X0",
          claimReference: 1188,
          claimStatus: "Closed",
          claimDate: "2024-11-15",
          statusLastUpdated: "2024-12-03"
        }
]
```

Invalid – example


```
{
   code: "404_40,
   text: "Policy not found",
   description: "Policy not found. Check Policy parameter.",
   technical: "",
   helpPageUrl: ""
}
```

Return the URL of the hosted Claims Portal

REQUEST

Method: GET

Staging URL: https://embapistaging.quotall.com/v1.0/claims/portal/{policyReference}

Production URL: https://embapi.quotall.com/v1.0/claims/portal/{policyReference}

Headers:

Name	Value
"Content-Type"	"application/json"
"Authorization"	"Bearer *token*"

Body: None

RESPONSE

Success – example


```
{
   url:
"https://subdomain.quotall.com?clr=YkE4dDdVcGhGdFNFRHxHQmFTQQ=="
}
```


Invalid – example


```
{
   code: "400",
   text: "Invalid input field",
   description: "Policy with that reference does not exist",
   technical: "",
   helpPageUrl: ""
}
```

Data Dictionary

Name	Data Type	Max Length	Description
answerType	string	n/a	The answer type of the claims question. This suggests which HTML element should be surfaced to collect the answer.
autoRenew	string	1	“on” or “off” determines whether or not this policy will renew automatically.
claimDate	date	n/a	The date the claim was made.
claimReference	integer	n/a	Returned by the claims/create endpoint.
claimStatus	string	n/a	The current status of the claim.
client_id	string	n/a	Supplied by Quotall. Forms part of the Api user orgaisation authentication credentials used to obtain an authentication token.
client_secret	string	n/a	Supplied by Quotall. Forms part of the Api user orgaisation authentication credentials used to obtain an authentication token.
customerDetails	object	n/a	An object with sub-properties.
customerDetails.address	object	n/a	An object with sub-properties.

customerDetails.address.addressLine1	string	40	The first line of your customer's address.
customerDetails.address.addressLine2	string	28	The second line of your customer's address.
customerDetails.address.postCode	string	40	The postal code of your customer's address.
customerDetails.address.townCity	string	40	The town or city of your customer's address.
customerDetails.customerId	string	1,000	Your organisation's unique customer id.
customerDetails.email	string	80	Your customer's email address.
customerDetails.firstName	string	30	Your customer's first name.
customerDetails.lastName	string	30	Your customer's last name.
grossPremium	decimal	n/a	Total premium paid by the customer including commission and IPT.
iptAmount	decimal	n/a	The Insurance Premium Tax amount included in the gross premium.
isMandatory	boolean	n/a	Whether or not this claims question must be answered.
options	object	n/a	An object with sub-properties.
options.code	string	n/a	The code to populate an HTML element with options such as a <select> element. This code should be returned in the claims/create endpoint.

options.text	string	n/a	The code to populate an HTML element with options such as a <select> element. This is the text that should be displayed to the customer.
orderItemId	string	1,000	Your organisation's unique Id for the item within an order.
orderReference	string	1,000	Your organisation's unique order reference for the order containing the insurance element (optional).
paymentReference	String	1,000	The payment reference for a renewal payment.
policyExpiryDate	date	n/a	The expiry date of the policy. The customer will be off cover on the day after this date unless they renew their policy.
policyReference	string	12	The Quotall policy reference as returned by the policies/create endpoint.
policyStatus	string	n/a	The status of a policy. Possible values: active cancelled expired
policyTypeId	string	n/a	The policyTypeId as returned by the policies/policyInformation endpoint.
productCategory	string	50	The category of the product to be insured. A list of categories

			must be supplied by your organisation.
productSubCategory	string	50	The subcategory of the product to be insured. This enables products with different IPT rates to co-exist in the same parent category. Optional.
productId	string	1,000	Your organisation's unique product id.
policyInformation	object	n/a	An object with sub-properties.
policyInformation.annualPrice	decimal	n/a	The annual premium for the insurance.
policyInformation.currency	string	n/a	The ISO 4217 alphabetic currency code for the prices.
policyInformation.monthlyPrice	decimal	n/a	The monthly premium for the insurance (the insurance is always annual).
policyInformation.policySummary	string	n/a	A brief summary of the insurance cover.
productName	string	1,000	The name of the product to be insured. Should include the brand and model.
productRetailPrice	decimal	n/a	The standard retail price of the item to be insured.
productSalePrice	decimal	n/a	The price paid for the item by the insured.
questionAnswers	object	n/a	An object with sub-properties.
questionAnswers.answer	string	5,000	The answer to a claims question.
questionAnswers.questionId	integer	n/a	The id of a claims question as returned by the

			claims/questions endpoint.
questionId	integer	n/a	The id of a claims question to be used as an input to the claims/create endpoint.
questionText	string	n/a	The text of a claims question to be surfaced to a customer making a claim.
refundAmount	decimal	n/a	The amount to refund on policy cancellation.
renewingPolicyReference	string	12	The policy reference of the renewing policy, for getting a price for renewals.
statusLastUpdated	date	n/a	The date of the most recent claim status update.
token	string	n/a	The JWT token to be used as a Authorization header in all other endpoints.
url	string	n/a	The URL of a hosted claims page to which to direct customers wishing to make a claim on their policy if you don't create your own claims page.

Appendix

Postman Collection

Data

Supplied on request.

Documentation

Documentation can be viewed here: https://apidocs.Quotall.co.uk

Amendments

v0.1	16/08/2024	Released.
v0.2	02/09/2024	Added Claim Document Upload.
v0.3	03/09/2024	Text amended.
v0.4	10/09/2024	Data Dictionary added.
v0.5	18/09/2024	Text changes.
v0.6	20/09/2024	Added Auto-Renew endpoint and renewals process description.
v0.7	25/09/2024	Endpoints updated.
v0.8	10/10/2024	Added new “Title” Claim Question answer type.
v0.9	16/10/2024	Add new endpoint customers/updateCustomer. Add new Claims Question Options and Claim Question Answer Types.
v1.0	04/12/2024	Removed endpoint claims/status/{claimReference}, replacing it with claims/status/{policyReference}. The replacement endpoint returns details of all claims on a policy rather than a single claim by claim reference.
v1.0.1	13/01/2025	Correct addressLine2 max characters to 28.
v1.0.2	11/02/2025	Add Order Reference to the request and response of the policies/create endpoint.
v1.0.3	13/03/2025	Add service availability endpoint
v1.0.4	15/04/2025	Added new document endpoint to retrieve IPID document.
v1.0.5	12/05/2025	Add optional sub-category item to policies/create and policies/policyInformation endpoints.
v1.0.6	10/11/2025	Add the mandatory field orderItemId to the policies/create endpoint.

v1.0.7	09/01/2026	Add the expiry date of the renewed policy to the response of the policies/renew endpoint.
v1.0.8	24/04/2026	Return the status of the policy to both policy endpoints customers/policy and customers/policies. This status can be used to decide which facilities will be available on your customer account page for each customer policy. The customers/policy endpoint only returns Accepted (not Cancelled) policies. Add a renewingPolicyReference property to the request of the policies/policyInformation endpoint. This is backwards compatible - leaving out or leaving blank the renewingPolicyReference means it will behave as before, so new business calls will not change, but for renewal price calls the renewingPolicyReference should be populated and the call will then also check that the policy is active. If it's active it'll return the price as now, if not it'll return an error.
v1.0.9	17/08/2026	Add auto-renew flag to response of customers/policy and customers/policies endpoints.
v1.1.0	29/09/2026	Return premium information with customers/policy and customers/policies endpoints for reconciliation purposes. Return IPT amount with polices/policyInformation endpoint for potential display to customers.

Quotall Limited is registered in England and Wales. Registered number: 7215557. VAT number: 991115323. Registered office: c/o Hillier Hopkins LLP, First Floor, Radius House, 51 Clarendon Road, Watford, Hertfordshire, United Kingdom, WD17 1HP. Quotall (IS) Limited is a wholly owned subsidiary of Quotall Limited and an Appointed Representative of Movo Partnership Ltd who are directly authorised and regulated by the Financial Conduct Authority, authorisation number 823503. Registered Office: First Floor, 30 High St, Chislehurst Kent BR7 5AS Registered in England No: 11544238. Quotall Insurance Marketing is a trading name of Quotall Limited.