vision

API Documentation V r1.0

LetAlliance

AMBER – Restricted

Contents

AMBER – Restricted

Document last updated: 21/02/2023 01:26:00

Change History

Version	Release Date	Description	Author
r1.0	01/10/2022	Initial Document Release	LW

Support Contacts

For additional support or any queries related to the Vision+ API, please email:

vp-apisupport@pib-insurance.com

Document Overview

This document contains details of the steps required for Let Alliance customers to integrate their web portals/software with the Vision+ system and provide seamless referencing functionality for their tenants and landlords.

Document Usage

This document is to be used as a guide and reference library for the following:

- Setting up and initial connection to Vision+ API
- For reference and understanding the functions provided
- Troubleshooting existing integrations at a functional level

The accompanying data dictionary and field map document is to be used for the following:

- Understanding field and object definitions
- Mapping of enumeration tables
- Troubleshooting existing integrations at an object level

The pre production Swagger API test website is used for:

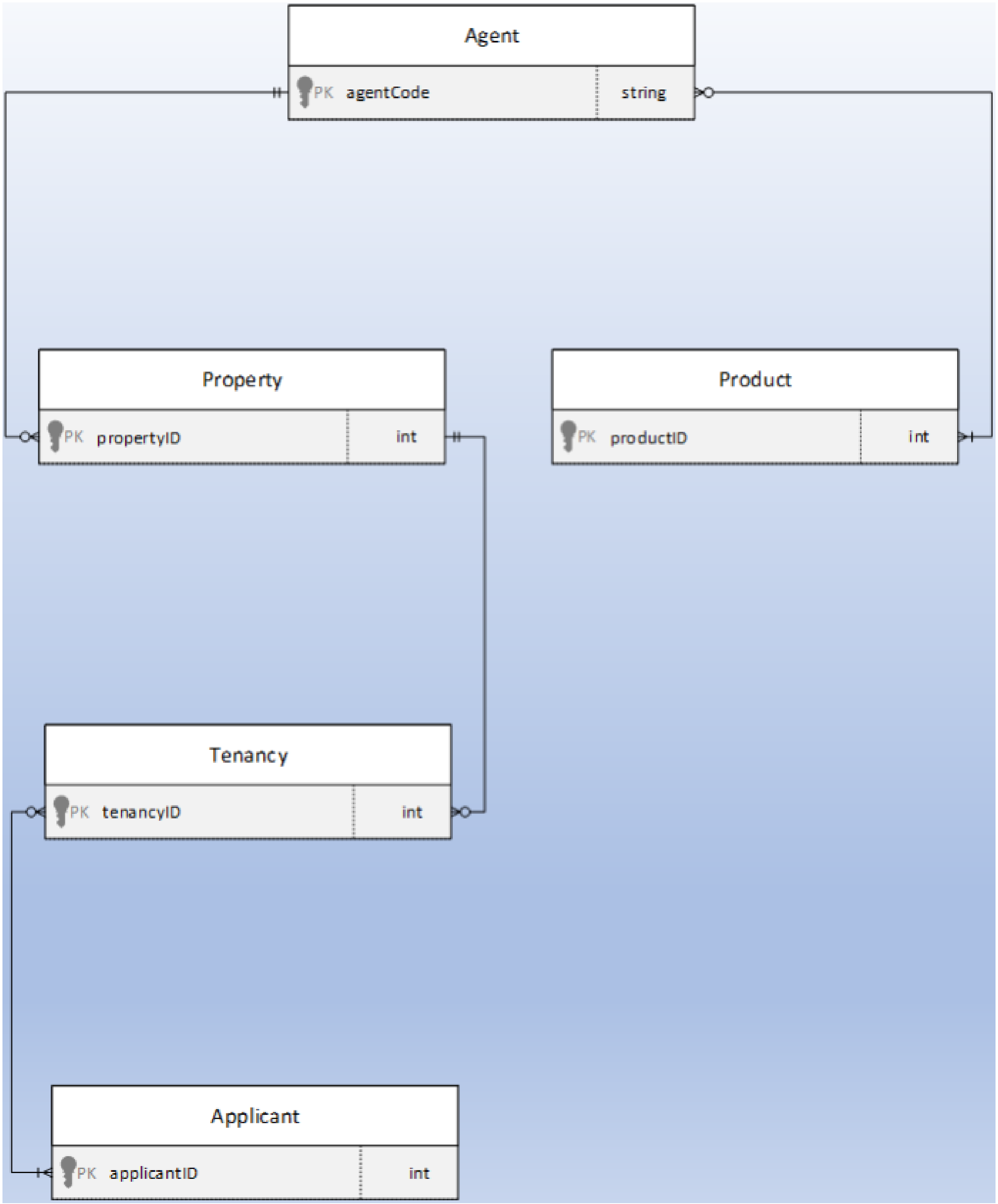
- Mapping of field data, with types and field lengths
- Testing and creation of JSON Payloads
- Troubleshooting existing integrations at a detail level

Vision+ API Functionality

The Vision+ API provides REST based API methods for:

- Authenticating
- Checking Connection
- Creating Tenant Reference
    - With all data from system
    - With input link sent to applicant
- Creating company Reference
    - With all data from system
    - With input link sent to applicant
- Checking Tenant Status
- Agent Product List
- Download Assessment Reports
- Upload Document
- Add notes
- Re send applicant link
- Amend applicant
- Add extra applicant
- Remove applicant

Entity Definition and Relationship



Agent

The agent object denotes the Vision+ customer, e.g. Estate Agent or branch if a group company

Product

Each Agent has a set of active/allowed products due to agreed commercial terms. These denote which products are available to use through the API such as RentOnTime or Ultimate reference. See enum Product list in supporting document for full listing, and use the GetAgentActiveProducts call to get the products active for the agent that is authenticated.

Property

A property is set up using any of the create reference calls, and is the base object for the house or dwelling that the reference application is for.

Tenancy

The tenancy object represents the overall reference object, linking to agent and property via their respective IDs.

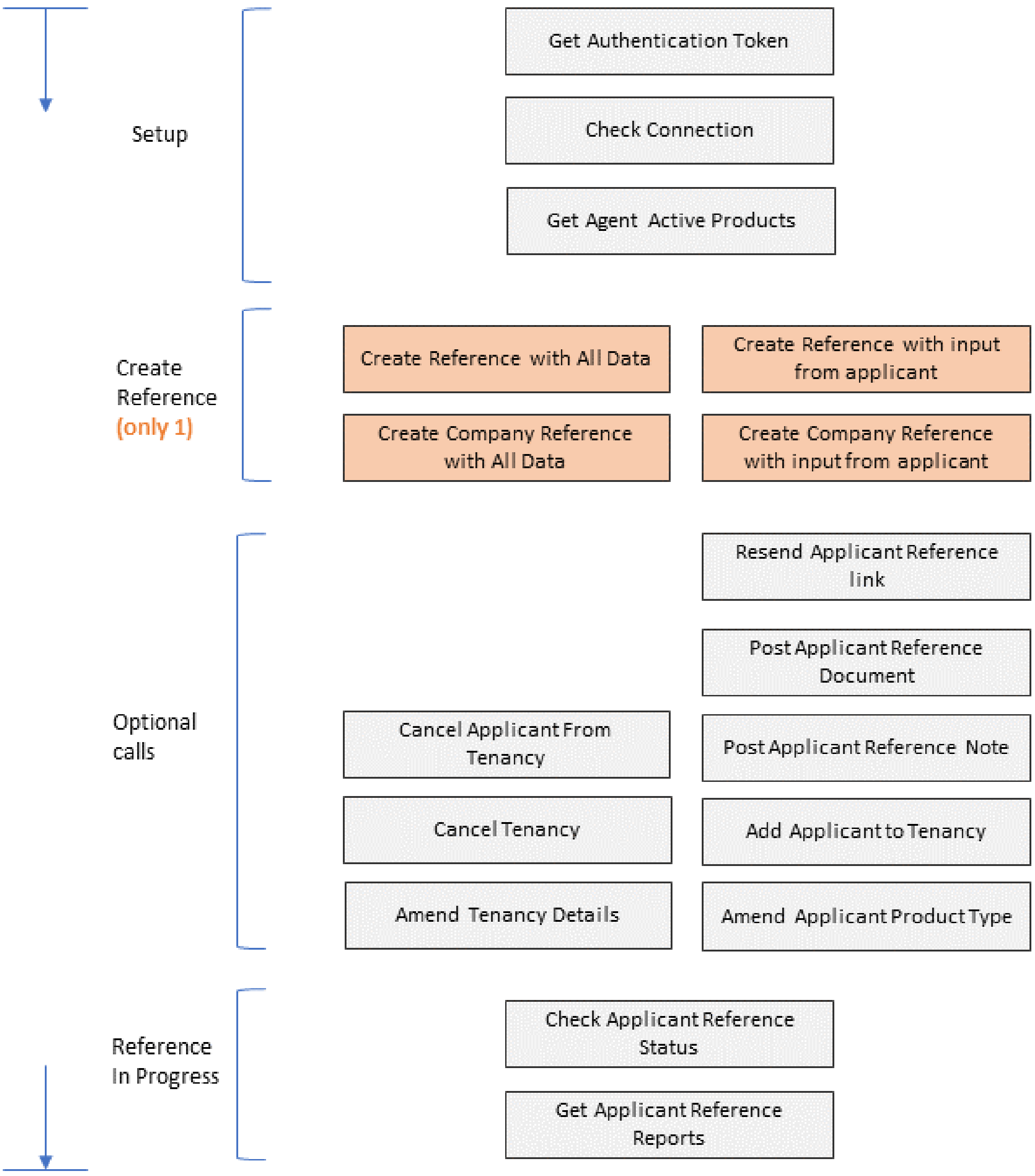
Applicant

As a child of the Tenancy object, the applicant represents the individual or company being referenced and contains all the relevant data. This data can either be provided by the system using the API, or alternatively an email containing a link can be sent to the applicant to allow them to enter the data directly into the Vision+ system themselves.

AMBER – Restricted

API Function Call Order

The API functions should be used in the following order, from top to bottom in the supporting diagram.


Setup
Get Authentication Token
Check Connection
Get Agent Active Products
Create Reference (only 1)
Create Reference with All Data
Create Reference with input from applicant
Create Company Reference with All Data
Create Company Reference with input from applicant
Optional calls
Resend Applicant Reference link
Post Applicant Reference Document
Cancel Applicant From Tenancy
Post Applicant Reference Note
Cancel Tenancy
Add Applicant to Tenancy
Amend Tenancy Details
Amend Applicant Product Type
Reference In Progress
Check Applicant Reference Status
Get Applicant Reference Reports

Integration Method

Vision+ API provides a staging and production end point, allowing testing to be performed in a safe sandbox environment prior to live integration.

Note: All API requests must be made over HTTPS. Calls made over standard HTTP or without authentication will fail.

Environment End Point

The end point will be shared with you as part of your agent credentials, which can be requested via the API setup form.

Authentication

Vision+ API use industry standard OAuth 2.0 for authentication of incoming requests and authorisation to API request. All client API request needs to have a valid bearer token generated from Authentication URL. Any request without Valid token will fail with HTTP Status code 403.

It is recommended clients use an API integration tool like Postman or Swagger to generate a bearer token and test API integration in the first instance. Examples can be provided to help on request.

Please co-ordinate with the Vision+ API onboarding team to get below security details for generation of Token.

- Access Token URL
- Client Id
- Client Secret
- Scope

Validity period of Token is 24 hours. For an example of the authentication call, see API Methods.

Request Response Formats

The Vision+ API accepts form-encoded request bodies, returns JSON-encoded responses, and uses standard HTTP response codes, authentication & commands.

Response Format

All the API has standard response format, as below:

```
{
  "success": true,
  "payload": {},
  "message": null
}
```

All API methods have three response parameters

- Success – Denotes if last API call was successful or not
- Payload – Contains the API call specific detail
- Message – Provides error messaging where required

Vision+ API status codes

The Vision+ API system uses standard HTTP codes to validate if the request was successful or not, for reference these are:

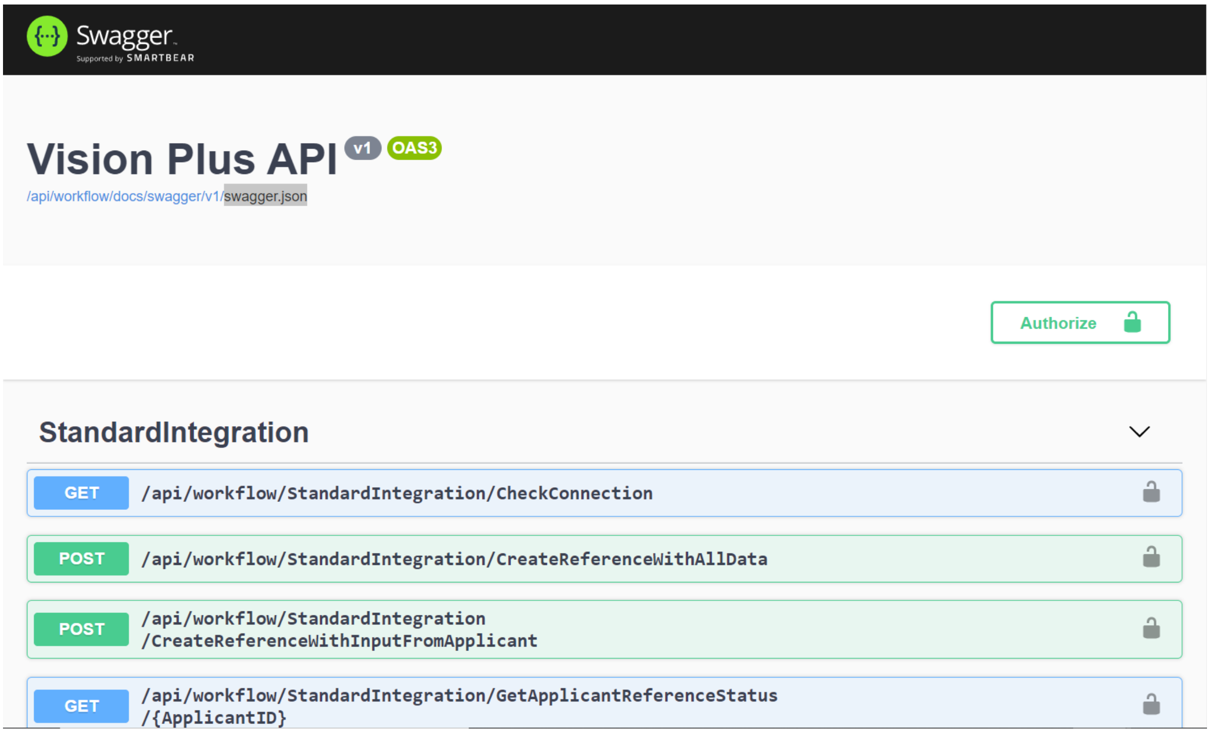
Code	Description
200	OK
201	Resource created.
401	Unauthorised request
403	Forbidden
404	Resource not found
422	Unprocessable entity or invalid syntax
500	Internal Server error

AMBER – Restricted

Swagger Test Environment

A Swagger test environment has been created to assist in testing and understanding the Vision+ API calls. This provides a functional connection into the pre production Vision+ API, with bearer token creatin, data definitions and example templates. To use the Swagger environment, the following steps should be followed:

Navigate to the Swagger URL : https://kr51a3q1x5.execute-api.eu-west-2.amazonaws.com/standardapi/docs/swagger/index.html



Vision Plus API v1 OAS3

/api/workflow/docs/swagger/v1/swagger.json

StandardIntegration

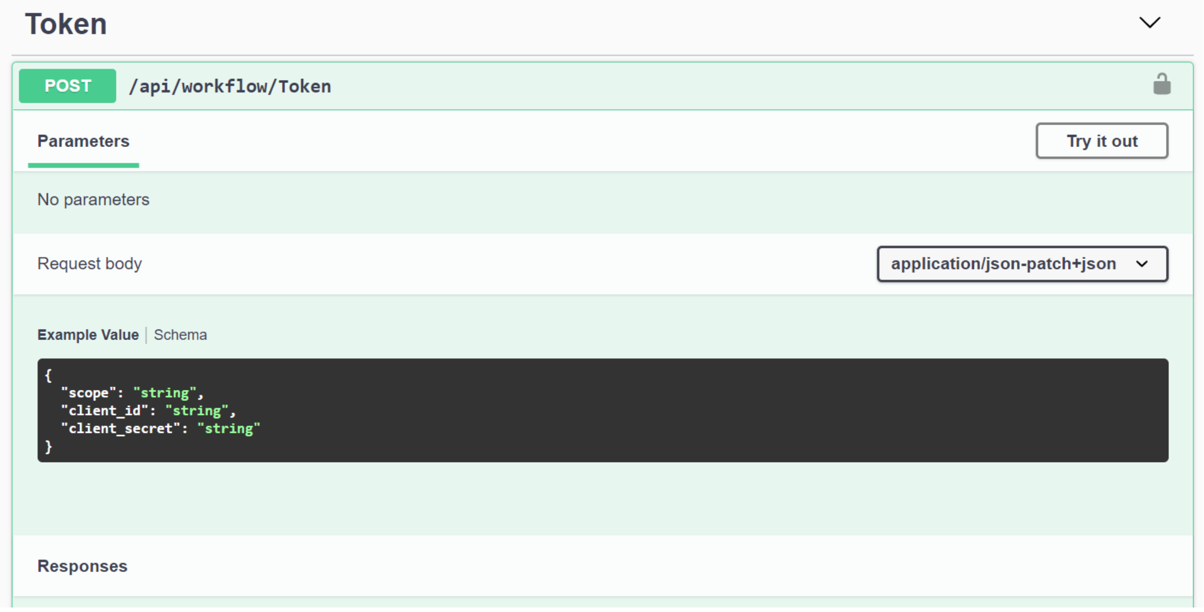
GET /api/workflow/StandardIntegration/CheckConnection

POST /api/workflow/StandardIntegration/CreateReferenceWithAllData

POST /api/workflow/StandardIntegration/CreateReferenceWithInputFromApplicant

GET /api/workflow/StandardIntegration/GetApplicantReferenceStatus/{ApplicantID}

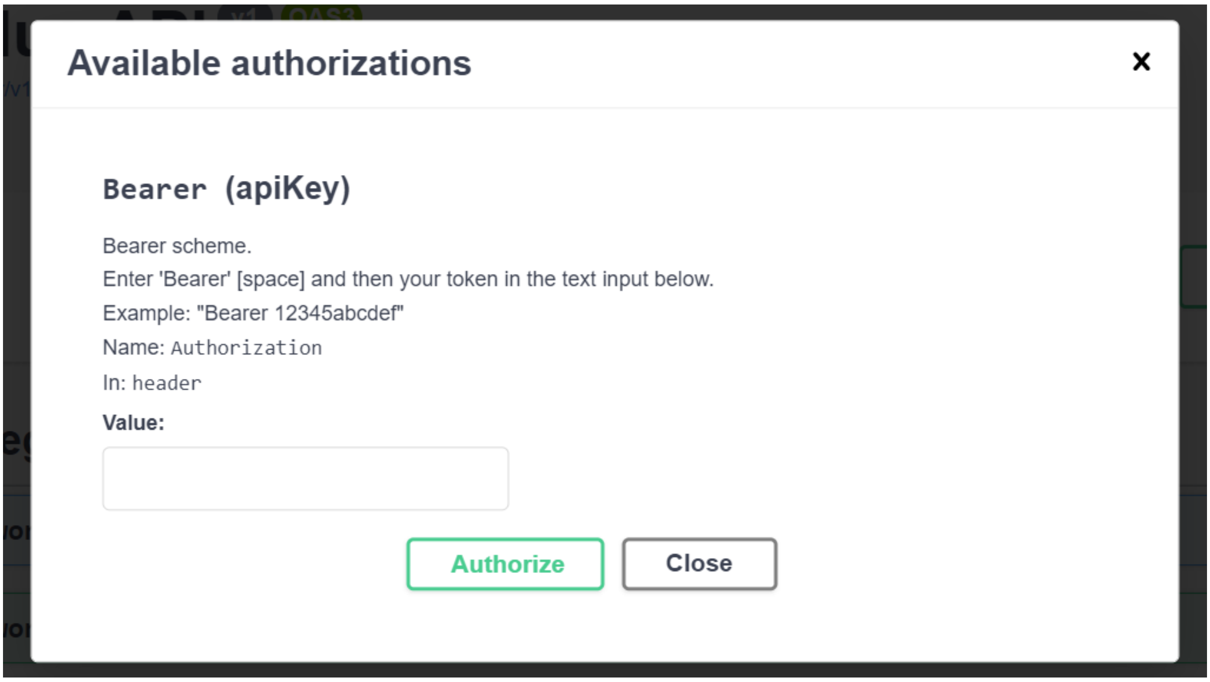
A list of all the available API calls will be displayed. To access these, first the authentication bearer token must be generated using the details provided.



```
{
  "scope": "string",
  "client_id": "string",
  "client_secret": "string"
}
```

Use the Try it now button to enter the details and execute to generate the token.

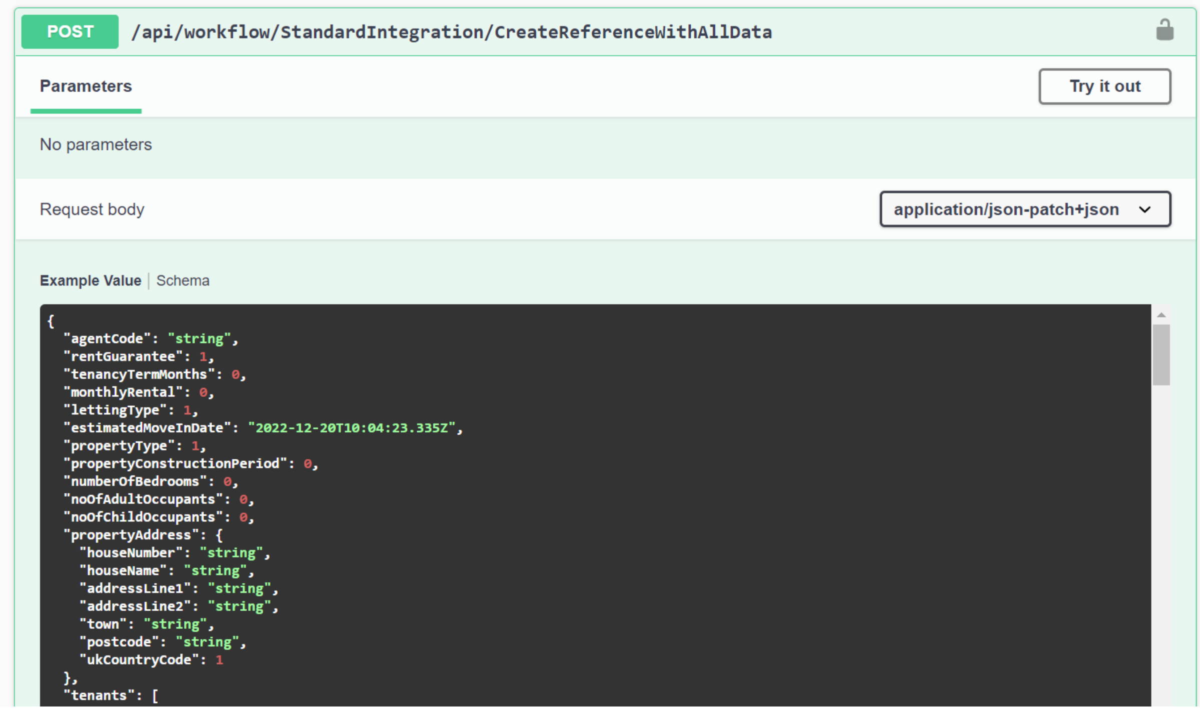
Then click the authorise button and paste the bearer token into the field, followed by Authorise.



Bearer scheme.
Enter 'Bearer' [space] and then your token in the text input below.
Example: "Bearer 12345abcdef"
Name: Authorization
In: header

Post authorisation, choose a method to test, in this case **CreateReferenceWithAllData**. An example JSON request will be shown.

AMBER – Restricted



/api/workflow/StandardIntegration/CreateReferenceWithAllData

Parameters

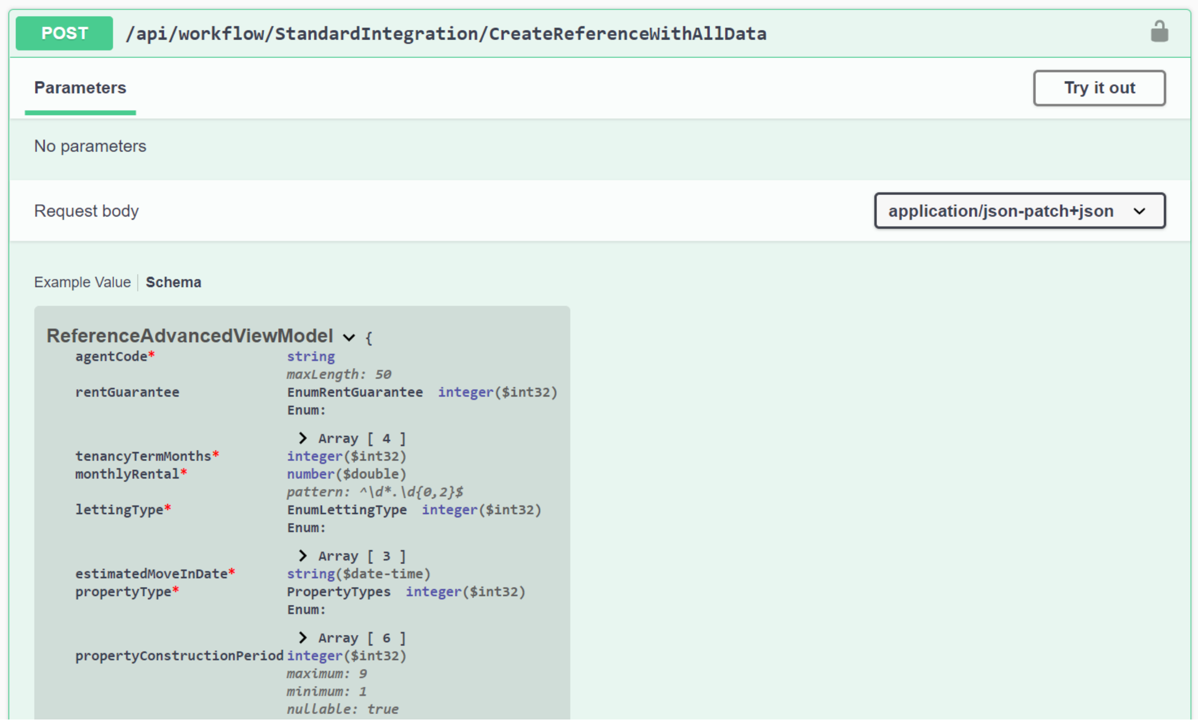
No parameters

Request body: application/json-patch+json

Example Value | Schema

```
{
  "agentCode": "string",
  "rentGuarantee": 1,
  "tenancyTermMonths": 0,
  "monthlyRental": 0,
  "lettingType": 1,
  "estimatedMoveInDate": "2022-12-20T10:04:23.335Z",
  "propertyType": 1,
  "propertyConstructionPeriod": 0,
  "numberOfBedrooms": 0,
  "noOfAdultOccupants": 0,
  "noOfChildOccupants": 0,
  "propertyAddress": {
    "houseNumber": "string",
    "houseName": "string",
    "addressLine1": "string",
    "addressLine2": "string",
    "town": "string",
    "postcode": "string",
    "ukCountryCode": 1
  },
  "tenants": [
```

Clicking on the schema tab will provide in depth field definition. This is further supported by the supplementary Data dictionary specification document



/api/workflow/StandardIntegration/CreateReferenceWithAllData

Parameters

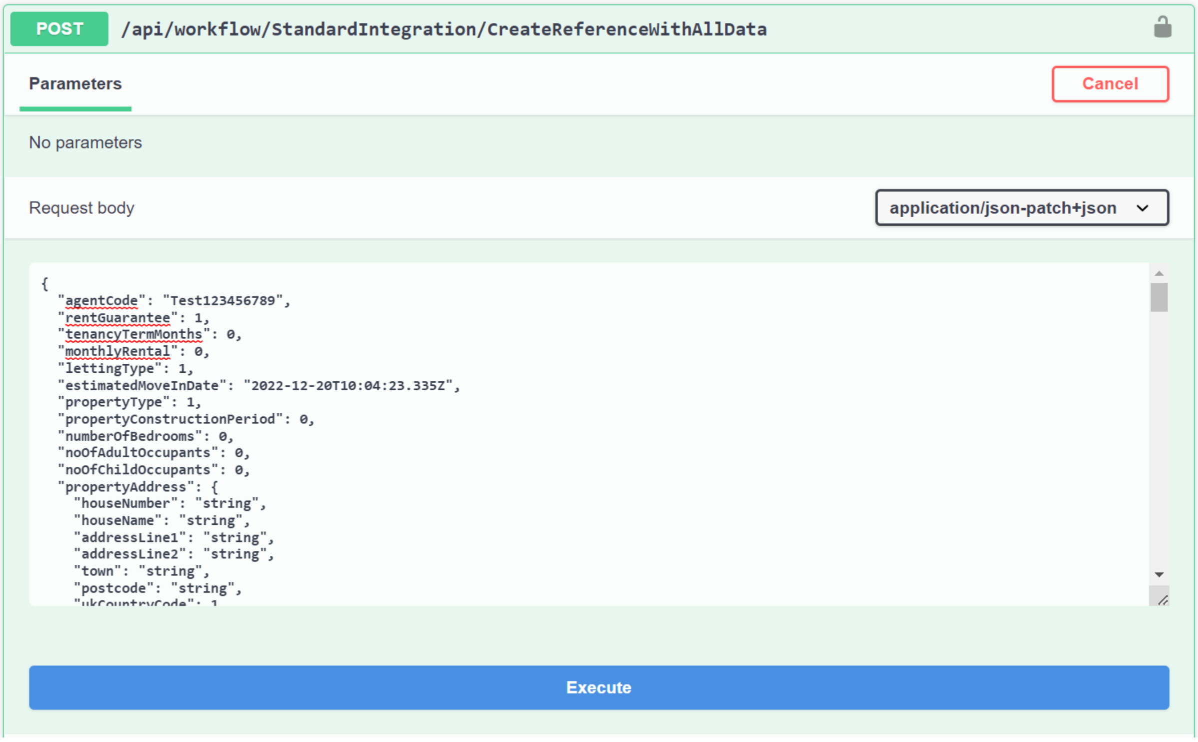
No parameters

Request body: application/json-patch+json

Example Value | Schema

```
ReferenceAdvancedViewModel ⌄ {
   agentCode*                 string
                              maxLength: 50
   rentGuarantee              EnumRentGuarantee  integer($int32)
                              Enum:
                                > Array [ 4 ]
   tenancyTermMonths*         integer($int32)
   monthlyRental*             number($double)
                              pattern: ^\d*.\d{0,2}$
   lettingType*               EnumLettingType  integer($int32)
                              Enum:
                                > Array [ 3 ]
   estimatedMoveInDate*       string($date-time)
   propertyType*              PropertyTypes  integer($int32)
                              Enum:
                                > Array [ 6 ]
   propertyConstructionPeriod integer($int32)
                              maximum: 9
                              minimum: 1
                              nullable: true
```

By clicking the try it now button, an editable fully formed example JSON is displayed. This should be edited to include the correct Agent ID and any field details that need to be amended.



/api/workflow/StandardIntegration/CreateReferenceWithAllData

No parameters

Request body: application/json-patch+json

```
{
  "agentCode": "Test123456789",
  "rentGuarantee": 1,
  "tenancyTermMonths": 0,
  "monthlyRental": 0,
  "lettingType": 1,
  "estimatedMoveInDate": "2022-12-20T10:04:23.335Z",
  "propertyType": 1,
  "propertyConstructionPeriod": 0,
  "numberOfBedrooms": 0,
  "noOfAdultOccupants": 0,
  "noOfChildOccupants": 0,
  "propertyAddress": {
    "houseNumber": "string",
    "houseName": "string",
    "addressLine1": "string",
    "addressLine2": "string",
    "town": "string",
    "postcode": "string",
    "ukCountryCode": 1
```

On Execute, the JSON payload will be processed by the pre production Vision+ API and the response will be displayed.

This tool can be used to test JSON created by bespoke systems and compare any responses.

API Methods

This section details each call with URL, method type and summary. Parameters, data descriptors and definitions are provided in the supplementary data dictionary documentation and the Swagger test environment.

Get Authentication Token

URL : <shared staging or production root URL> /Token

HTTP Method: GET

Summary: An initial authentication method used to generate a bearer token for the Vision+ API.

The call uses the credentials of client ID and client secret that will have been provided to you. The result should be used stored as an oAuth2 token with the name: [bearer] to authenticate any other API calls. The Token has a timeout of 24 hours before needing to be regenerated.

Note, the scope parameter should always be the constant value: “standardapiaccess".

Check Connection

URL : <shared staging or production root URL> /CheckConnection

HTTP Method: GET

Summary: Simple ping method used to check connectivity between Vision+ API server and client. As with all methods, if authentication token is missing or invalid, an error will be returned. If the authentication token is valid and the connection is successful, a success true will be returned.

Get Agent Active Products

URL: <shared staging or production root URL>/GetAgentActiveProducts

HTTPMethod: GET

Summary: This method is used to get the list of products currently available to the agent to use in referencing applications. Using an invalid product code or a product the agent is not authorised for will cause the reference application request to fail and return an error.

Create Reference with All Data

URL : <shared staging or production root URL> / CreateReferenceWithAllData

HTTP Method: POST

Summary: As one of the base methods of the Vision+ API, this method is used to create a property, create tenancy on property and add one or more applicants to be referenced in that property, **for systems that gather and provide all required applicant data up front.**

The method returns property tenancy ID and any applicant IDs that have been created for the referencing application.

Create Reference with Input From Applicant

URL: <staging or production root URL>/CreateReferenceWithInputFromApplicant

HTTP Method: POST

Summary: As another base methods of the Vision+ API, this method is used to create a property, create tenancy on property and add one or more applicants to be referenced in that property, **for systems that require the applicant to fill in their own data.**

The method returns property tenancy ID and any applicant IDs that have been created for the referencing application.

On submission, an email will be sent to any Applicant in the application, with details on how to log on to the Vision+ Portal and complete the required referencing data.

Resend Applicant Reference Link

URL: <shared staging or production root URL> /ResendApplicantReferenceLink

HTTPMethod: GET

Summary: Used in conjunction with the references created with the CreateReferenceWithInputFromApplicant call, This method is used to resend the applicant referencing link to the applicant, to complete the required referencing data. This is not required for reference applications started with the CreateReferenceWithAllData call.

Get Applicant Reference Status

URL: <shared staging or production root URL>/ GetApplicantReferenceStatus

HTTP Method: GET

Summary: This method is used to check the Applicant referencing status of one or more Applicants. For each applicant, it will return one of the following status':

- In progress
- On hold (More data required)
- Completed
- Invalid

An invalid response will be shown for any applicant ID that is not an active applicant attached to the authenticated agent.

Get Applicant Assessment Reports

URL: <shared staging or production root URL>/ GetApplicantReferenceReports/{ApplicantId}

HTTPMethod: GET

Summary: This method provides the applicant available assessment reports as a base64 encoded string, which must be saved as PDF file. An interim report will be available if the appliants referencing application is in progress, and a final report will be avaible when the applicants reference application status is completed.

Post Applicant Reference Document

URL: <shared staging or production root URL> /PostApplicantReferenceDocument

HTTPMethod: POST

Summary: This method is used to upload documents to support the applicant assessment process. Such as identification, proof of income, rent receipts etc.

This method adheres to the same file size limitation (10 Mb) and acceptable file types (PDF, Doc*, XLS*, PNG, JP*G) as the Vision+ Portal.

Post Applicant Reference Note

URL: <shared staging or production root URL> /PostApplicantReferenceNote

HTTPMethod: POST

Summary: This method is used to add notes to the applicant assessment. The note field supports free text and can be used to record additional data.

Add Applicant to Tenancy

URL: <shared staging or production root URL>/ AddApplicantToTenancy

HTTPMethod: POST

Summary: This method is used to add an additional applicant to a tenancy application. It has an applicant type parameter that denotes whether the applicant will be sent a link to fill out the data themselves or that the data will be provided by the client system.

Cancel Applicant from Tenancy

URL: <shared staging or production root URL>/ CancelApplicantFromTenancy

HTTPMethod: POST

Summary: This method is used to get cancel a specific applicant from a tenancy application associated with the authenticated agent.

Cancel Tenancy

URL: <shared staging or production root URL>/ CancelTenancy

HTTPMethod: POST

Summary: This method is used to cancel a tenancy application associated with the authenticated agent.

Amend Tenancy Details

URL: <shared staging or production root URL>/ AmendTenancyDetails

HTTPMethod: POST

Summary: This method is used to amend a tenancy referencing application associated with the authenticated agent.

Amend Applicant Product Type

URL: <shared staging or production root URL>/ AmendApplicantProductType

HTTPMethod: POST

Summary: This method is used to amend the product type used for a specific applicant. This is restricted by the current agent active products and only for applicants associated with the authenticated agent.

Create Company Reference with All Data

URL: <shared staging or production root URL>/ CreateCompanyReferenceWithAllData

HTTPMethod: POST

Summary: As one of the base methods of the Vision+ API, this method is used to create a company reference with a property, create tenancy on property and add one or more applicants to be referenced in that property, **for systems that gather and provide all required company data up front.**

The method returns property tenancy ID and any applicant IDs that have been created for the referencing application.

Create Company Reference with Input From Applicant

URL: < staging or production root URL>/ CreateCompanyReferenceWithInputFromApplicant

HTTPMethod: POST

Summary: As another base methods of the Vision+ API, this method is used to create a company reference with a property, create tenancy on property and add one or more applicants to be referenced in that property, **for systems that require the applicant to fill in their own data.**

The method returns property tenancy ID and any applicant IDs that have been created for the referencing application.

On submission, an email will be sent to any Applicant in the application, with details on how to log on to the Vision+ Portal and complete the required referencing data.