User Guide:
Client Feed Export (API)

Document Creation Date:	07 June 2024
Document Version:	13.1
API version:	13

Document History

Date *(dd mmm yyyy)*	*Version* *(n.nn)*	*Reason for Change* *(including paragraph references)*
07 June 2024	13.1	**First Issue for v13 Schema changes:** 1. Current version is v13 2. Schema links updated to reflect current version 3. All examples of URLs updated to reflect current version The primary reason for this release is to make additional changes required by national trading standards. **Schema changes in this version:** **Propertyexport.xsd** 1. Addition of new **negotiator** element 2. Addition of new **features** element 3. Addition of the following elements to the new **features** element: - **accessibility_requirements** - **broadband** - **building_safety** - **construction** - **coastal_erosion** - **electricity** - **flooding_risks** - **heating** - **known_planning_considerations** - **mining_risks** - **mobile_coverage** - **parking** - **restrictions** - **rights_and_easements** - **sewerage** - **water** **Propertyrelatedtypes.xsd** 1. Addition of the following types: - **ElectricitySupply** - **WaterSupply** - **SewerageSupply** - **HeatingSource**

Date *(dd mmm yyyy)*	*Version* *(n.nn)*	*Reason for Change* *(including paragraph references)*
		- BroadbandSupply - **ParkingType** - **Accessibility**

Table of Contents

Purpose of Document

1. The purpose of this document is to provide a technical user guide to The Property Software Group's Client Feed Export (API).
2. Although this document contains technical information, all attempts have been made to use plain English throughout.
3. The intended audience of this document are clients of The Property Software Group and their web developers.
4. The accompanying PHP script is provided as an example of consuming the API in PHP. We do not and cannot provide technical support on implementing a PHP solution to developers.

Summary of API Functionality

The Client Feed Export (API) provides a method by which clients of The Property Software Group are able to retrieve their data in real time in a standardized format so it can be stored in your database to power your website. This is delivered via a web XML API.

We will support multiple versions of the API. We will notify you when new versions of the API become available and/or when versions are due to be deprecated.

Your data feed identifier

You have been given a data feed identifier. This identifier is associated with the branch or branches for which you will be retrieving data. You will use this identifier in all API calls to the service where you see {datafeedid} in the URL. Use your identifier in place of {datafeedid} with no whitespace before or after it. The data feed identifier is not case sensitive

Authentication and Use of Tokens

API calls must be authenticated using HTTP Basic Authentication.

You have been given a username and password for your data feed - these are case sensitive. You will need to add an 'Authorization' header to the request, containing the string "username:password" (encoded to base 64), with the word "Basic " (plus space) in front.

The response to your initial request will contain a security token (accessed by the name 'Token' in the header) which must replace the 'username:password' string in subsequent requests. The token is valid for one hour and **must** be used until it expires – when it does, a 401 (unauthorized) will be returned and another token must be requested, using the username & password again.

C# Asp.net Example 1:

```
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

string username = "user";
string password = "password";

string usernamePassword = username + ":" + password;
```

```
CredentialCache cache = new CredentialCache();

cache.Add(new Uri(url), "Basic", new NetworkCredential(username,
 password));
request.Credentials = cache;
request.Headers.Add("Authorization", "Basic " // <- space here.
 + Convert.ToBase64String(new ASCIIEncoding().GetBytes(usernamePassword)));

// Get the token from the response:
string token = response.GetResponseHeader("Token");
```

C# Asp.net Example 2 - using the token in subsequent requests:

```
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

string token = "token1";

CredentialCache cache = new CredentialCache();

cache.Add(new Uri(url), "Basic", new NetworkCredential(username,
 password));
request.Credentials = cache;
// Use token here instead of username:password string.
request.Headers.Add("Authorization", "Basic " // <- space here.
 + Convert.ToBase64String(new ASCIIEncoding().GetBytes(token)));
```

API Calls

There are six API calls.

- Get Branches – returns a list of all branches
- Get Branch – returns more details for a branch
- Get Property List – returns list of properties for a branch
- Get Property – returns the property details
- Get Changed Properties – returns a list of properties that have changed (updated, added or deleted) since a certain date and time.
- Get Changed Files – returns a list of files that have changed (updated, added or deleted) since a certain date and time.

These API calls must all be authenticated.

All the API calls return XML. This XML is defined by a schema. This schema documents the XML returned.

API Versions

The Client Feed Export (API) supports multiple versions of the xml and appropriate schema, allowing The Property Software Group to add data and functionality to the API without affecting existing users.

We will notify you when new versions of the Client Feed Export (API) are available.

In line with Google, Twitter and Facebook, older versions of the API will be deprecated in time and sufficient notice will be given to migrate to the latest version.

The Client Feed Export (API) version is part of the URL on all API calls. All API URLs start with

https://webservices.vebra.com/export/{datafeedid}/v{version}

{version} is the version required. The current version is 13

Encoding

Previous versions of the Client Feed Export (API) were encoded in ISO-8859-1. Additionally, HTML escape codes were used for three characters not supported by ISO-8859-1, namely

1. Pound Symbol (£)
    a. Escape code = £
2. Euro Symbol (€)
    a. Escape code = €
3. Superscript to denote Square Feet / Metres (²)
    a. Escape code = ²

From API version v8, the Client Feed Export (API) is now encoded in UTF-8.

However, to assist existing users to upgrade to this version with minimal impact, the above characters will continue to be returned in the xml using the HTML escape codes as detailed above.

Please refer to Appendix 4 – HTML Escape Code Support for details of elements and attributes for which the HTML escape codes are returned in the Get Property call.

Caching

The Branch, Branches, Property and Property List URLs implement the Last-Modified header to allow caching by the client. If the client sets the If-Modified-Since header in the request, only records modified since that date and time will be returned.

Note 1: How the if-modified-header option works with the properties list differs to the way the properties changed call works:

When the if-modified-header option is used, the API gets the most recent property changed value for the client and compares it to the if-modified value. If the if-modified value is earlier, the API returns **all properties; otherwise none**.

The Get Changed Properties call only returns properties where the last change for the record is after the date specified.

Schema

Each version of the API will have a corresponding schema and the XML shall reflect this.

For example;

API version 13: *https://webservices.vebra.com/export/xsd/v13/exportapi.xsd*

The XML returned will include the namespace in the root node, e.g.

<branch xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="https://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation="https://webservices.vebra.com/export/xsd/v13/ exportapi.xsd">

The following schema files are also referenced:

Property Export: *https://webservices.vebra.com/export/xsd/v13/propertyexport.xsd*

Currency: *https://webservices.vebra.com/export/xsd/v13/currency.xsd*

Country, Market and Category: *https://webservices.vebra.com/export/xsd/v13/dbids.xsd*

Property Related Types: *https://webservices.vebra.com/export/xsd/v13/propertyrelatedtypes.xsd*

Rightmove Types: *https://webservices.vebra.com/export/xsd/v13/rightmovetypes.xsd*

Simple Types: *https://webservices.vebra.com/export/xsd/v13/simpletypes.xsd*

User Guide

The latest API v13 User Guide (this document)
https://webservices.vebra.com/export/xsd/v13/Client_Feed_API_v13_UserGuide.pdf

How to use the API

This section provides guidance on how you would use the API to initially populate your database and keep your data in sync, in near to real-time.

Initial Population of your database

1. Call 'Get Branches' – this returns XML with a list of branches and a URL to the branch details for each branch.

2. For each branch in that list, call 'Get Branch' and store the branch details in your database. (It would be useful to store the URL and date and time you queried this branch. Then you can periodically check this and use the date and time in the 'If-Modified-Since' header. If the data has changed you will get the changed data, otherwise the API will return 304 Not-modified. This can help you reduce the volume of database updates.)

3. For each branch in that list, call 'Get Property List'

4. For each property that 'Get Property List' returns, call 'Get Property' (the URL returned by 'Get Property List') and store the properties locally.

5. For each property you should download the image(s) and referenced file(s) and store them.

6. It would be sensible to allow your system to be able repeat the 'initial population of your database' if you lose sync, or in the event we notify you that this is required due to changes we have made.

Note 1: How the if-modified-header option works with the properties list differs to the way the properties changed call works:

When the if-modified-header option is used, the API gets the most recent property changed value for the client and compares it to the if-modified value. If the if-modified value is earlier, the API returns **all properties; otherwise none**.

The Get Changed Properties call only returns properties where the last change for the record is after the date specified.

On-going Real-time updates

Periodically you should use the API Calls detailed below to get changed properties and files. To do this:-

1. Call ‘Get Changed Properties’ using the date and time you last checked for updates / initially populated your database.
2. For each property returned, perform the appropriate operation (get property then update, or insert or remove the property) to your database.
3. Store the date and time you checked for updates for the next time you call the API.
4. Repeat the same process for files using ‘Get Changed Files’

API Call List

This section details the list of calls available.

Get Branches

https://webservices.vebra.com/export/{datafeedid}/v{version}/branch

{datafeedid} is the identifier for this data feed.
{version} the current version is 13.

Returns XML with list of branches for the data feed containing

- branch name
- firmid
- branchid
- URL to full branch XML ('Get Branch' call)

Please see the schema for full details.

Example URL:

https://webservices.vebra.com/export/homeestates/v13/branch
(This URL is not active)

Get Branch

https://webservices.vebra.com/export/{datafeedid}/ v{version}/branch/{clientid}

{datafeedid} is the identifier for this data feed.
{version} the current version is 13.
{clientid} is the branch identifier.

This URL is present in the 'Get Branches' XML for each branch.

Please see the schema for full details of the XML returned.

Example URL:

https://webservices.vebra.com/export/ homeestates/v13/branch/1234
(This URL is not active)

Get Property List

https://webservices.vebra.com/export/{datafeedid}/ v{version}/branch/{clientid}/property

{datafeedid} is the identifier for this data feed.
{version} the current version is 13.
{clientid} is the branch identifier.

Returns XML with the list of properties for the branch specified in the {clientid} containing (for each property)

- prop_id (the property identifier)
- last changed date and
- URL to full property details XML ('Get Property' call).

Please see the schema for full details of the XML returned.

Example URL:

https://webservices.vebra.com/export/homeestates/v13/branch/1234/property
(This URL is not active)

Note 1: The Get Property List call returns results for current properties only ordered by updated date. This value is the last time the property, rooms or associated files changed.

Get Property

https://webservices.vebra.com/export/{datafeedid}/ v{version}/branch/{clientid}/property/{prop_id}

{datafeedid} is the identifier for this data feed.
{version} the current version is 13.
{clientid} is the branch identifier.
{prop_id} is the unique property identifier.

Returns XML with the full property details for the property specified in {prop_id}

Please see the schema for full details of the XML returned.

Example URL:

https://webservices.vebra.com/export/homeestates/v13/branch/1234/property/12345678
(This URL is not active)

Note 1: The updated value for the files in the Get Property call will return the latest update value we hold where either the file itself or the media content has changed

Note 2: The Get Property call includes entries for EPC Graphs where the property EPC values are non-zero and we do not hold associated EPC files for the property record. These file entries are 'generated' by the API itself and will not contain the 'updated' attribute, nor will they be included in the Get Changed Files call.

- The individual Energy efficiency chart where the corresponding **eecurrent *and* eepotential** values are greater than 0
- The individual Environmental impact chart where the corresponding **eicurrent *and* eipotential** values are greater than 0

Get Changed Properties

https://webservices.vebra.com/export/{datafeedid}/v{version}/property/{yyyy}/{MM}/{dd}/{HH}/{mm}/{ss}

{datafeedid} is the identifier for this data feed.
{version} the current version is 13.

This call expects a date and time after which property changes are returned.

- {yyyy} – year
- {MM} – month with leading zeros
- {dd} – day of the month with leading zeros
- {HH} – the hour of the day (0 – 23) with leading zeros
- {mm} – minutes with leading zeros
- {ss} – seconds with leading zeros

Returns XML with a list of properties for the data feed that have been added, updated or deleted since the date and time specified, containing (for each property)

- prop_id
- last changed date
- last action on property, and
- URL to full property XML

Valid Values for Last Action on property are:

- updated (this refers to a new property record or an update to an existing property record)
- deleted

Example URL:

https://webservices.vebra.com/export/homeestates/v13/property/2010/08/31/01/01/01

(This URL is not active)

Note 1: The Get Changed Properties call returns results where the property record or its associated files or rooms (paragraphs in the xml) have changed with a date/time equal to or greater than the date / time in the call. It returns results where:

1. a new property record has been added
2. an update to an existing property record has occurred
3. an existing property has been deleted
4. the associated files have been updated or deleted

5. the associated rooms (paragraphs) have been updated or deleted
6. Additionally, if the Branch's Lettings Fee data has been updated, in this case, all properties will be returned for the Branch with the ***lastchanged*** date being the latest of the above changes and not the date the Lettings Fee data was added or updated

The ***lastchanged*** value in the returned results will always be the latest of all changes the call takes into consideration.

Note 2: The Get Changed Properties call returns results for current property records (last action = update) where a new property record has been added or is an update to an existing property record; or a deletion of an existing property record (last action = deleted).

Get Changed Files

https://webservices.vebra.com/export/{datafeedid}/v{version}/
files/{yyyy}/{MM}/{dd}/{HH}/{mm}/{ss}

{datafeedid} is the identifier for this data feed.
{version} the current version is 13.

This call expects a date and time after which property changes are returned.

- {yyyy} – year
- {MM} – month with leading zeros
- {dd} – day of the month with leading zeros
- {HH} – the hour of the day (0 – 23) with leading zeros
- {mm} – minutes with leading zeros
- {ss} – seconds with leading zeros

Returns XML with list of changed files for the feed containing

- file element with file_id
- file element with file_propid,
- last changed date
- a flag to denote if the file was deleted. If false, it was either added or updated.
- URL to full property XML ('Get Property' call)
- URL of the file.

Example URL:

https://webservices.vebra.com/export/homeestates/v13/files/2010/08/31/14/23/01
(This URL is not active)

Note 1: Deleted files are supported in this version of the API and the Get Changed Files call returns both current and deleted files. The <deleted> node in the xml returns a value of "false" for current files and "true" for deleted files.

Note 2: The Get Changed Files call includes the File Id of the changed files. When combined with the file_propid, this is the unique identifier for this file, allowing matching against previously downloaded files.

HTTP Status codes

The following status codes will be returned as appropriate:

Status	*Description*
200 – OK	Returning the XML document
204 – No Content	Returned if the branch for which the Get Property List request was made has no properties
304 – Not Modified	The 'If-modified-since' header was supplied and content has not been changed since **or** If the if-modified header is not used and no properties or have been updated since the date/time supplied in the Get Changed Properties request **or** No files have been updated since the date/time supplied in the Get Changed Files request
401 – Unauthorized	Returned if the authentication fails
403 – Forbidden	Returned if the client is authenticated but tries to access a URL that is not available to the client (i.e. another data feed, or property that is not in the your data feed)
404 – Not Found	Returned for an incorrectly formed URL
410 – Gone	The property/branch has been deleted OR is not currently available.
500 – Internal Server Error	Something bad happened to the API.
501 – Not Implemented	Returned if the API version in the request is not supported (deprecated or unreleased versions)

Glossary of Terms

Term or Abbreviation	*Meaning*
API	**Application Programming Interface** A description of the way one piece of software asks another program to perform a service.
HTTP	**Hypertext Transfer Protocol** A networking protocol for distributed, collaborative, hypermedia information systems.
PHP	**Hypertext Preprocessor** Server-side HTML embedded scripting language.
URL	**Uniform Resource Locator** A pointer to a "resource" on the World Wide Web. A resource can be something as simple as a file or a directory, or it can be a reference to a more complicated object, such as a query to a database or to a search engine.
XML	**Extensible Markup Language** A set of rules for encoding documents in machine-readable form

Appendix 1 – EPC Image Files

The Get Property call includes entries for EPC Graphs where the property EPC values are non-zero and we do not hold associated EPC files for the property record. These file entries are 'generated' by the API itself and will not contain the 'updated' attribute, nor will they be included in the Get Changed Files call.

- The individual Energy efficiency chart where the corresponding **eecurrent *and* eepotential** values are greater than 0
- The individual Environmental impact chart where the corresponding **eicurrent *and* eipotential** values are greater than 0

Where EPC image files are generated and referenced in the Get Property call (where the property has non-zero EPC values and we do not have associated EPC files for the property) they are branded for England & Wales, Northern Ireland or Scotland based on postcode as follows:

- If the Postcode starts **BT** the graphs generated will be branded as **Northern Ireland**
- If the Postcode starts with any of **AB, DD, DG, EH, FK, G, HS, IV, KA, KW, KY, ML, PA, PH, TD, ZE** the graphs generated will be branded as **Scotland**
- If the Postcode starts with anything else, the graphs generated will be branded as **England & Wales**

Appendix 2 – Properties Excluded from the Export

The functionality to allow an agent to exclude properties from the Export works as detailed below. This applies to all versions of the API.

Results for calls where property is not excluded

API Call	*Result*
Get Property List	The property will be included in the results
Get Property	The full details of the property will be returned
Get Changed Properties	The property will be included in the results with action = updated
Get Changed Files	Files for the property will be included in the results

Results for calls where property is excluded

API Call	*Result*
Get Property List	The property will not be included in the results
Get Property	Any attempt to retrieve the property details after the 2nd upload will result in a 404 Not Found response
Get Changed Properties	The property will be included in the results with action = deleted
Get Changed Files	No files for the excluded property will be included in the results In the event that the only files that have an updated value greater than the timestamp set in the call relate to the excluded property a 304 (Not Modified) response will be returned

Appendix 3 – Schema and Main Changes History

The following table details the schema changes and other main changes introduced in each API version released after the initial v1 API. Note that no v2 of the API was implemented.

API version	*Schema Change*
v3	**Schema changes in this version:** 1. Flag 'Featured' properties 2. Return large images 3. Include a flag for deleted files in the Get Changed Files call 4. LonRes fields included in Get Property call a. Solddate b. Leaseend c. Instructed d. Soldprice e. Garden f. Parking g. Groundrent h. Commission i. These will either return a value as per the propertyexport.xsd annotation or where we hold a NULL value, nil="true"
v4	**Schema changes in this version:** 1. Add file_id in Get Changed Files call 2. Annotation re date formats in the Propertyexport.xsd updated to give correct format where present and to include the date format where this was not previously documented 3. Schema changes to the propertyexport.xsd: a. Commercial section added b. Added optional "display" attribute to the property price element c. Added acre and hectare unit attribute on area d. Measure attribute on area made optional i. The area can be distinguished by the unit attribute which will show one of Sqft / Sqm / Hectare / Acre e. Area maxOccurs="2" removed f. Min, max elements in area changed to decimal g. Comm_rent deprecated h. Rm_type changed to xs:int i. Transaction node within the Commercial element has been made optional in the propertyexport.xsd

API version	*Schema Change*
v5	**Schema changes to the propertyexport.xsd in this version:** 1. Commercial section added 2. Added optional “display” attribute to the property price element 3. Added acre and hectare unit attribute on area 4. Measure attribute on area made optional 5. Area maxOccurs=”2” removed 6. Min, max elements in area changed to decimal 7. Comm_rent deprecated 8. Rm_type changed to xs:int 9. Expanded field lengths for various fields to cater for conversion of currency symbols (£ and €) and superscript ^{2} (square feet / metres etc.) a) To clarify, the area the propertyexport.xsd can be distinguished by the unit attribute which will show one of Sqft / Sqm / Hectare / Acre (relates to Point 1d) above)
v6	**Schema changes in this version:** 1. New Streetview element added to the propertyexport.xsd 2. New Landarea element added to the propertyexport.xsd 3. The ‘Updated’ or ‘LastChanged’ date formats in the Property and Files calls have been changed to have consistent formats of yyyy-mm-ddThh:mm:ss and impacts the following: a. the lastchanged attribute in the Get Property List call b. the lastchanged attribute in the Get Changed Properties call c. the files/updated attribute in the Get Property call d. the files/updated attribute in the Get Changed Files call **Other Changes:** Functionality added to allow properties to be excluded from the export. Appendix 2 added to provide details of how this will impact the Property and Files calls
v7	**Schema changes in this version:** 1. New Rentalfees element added to the propertyexport.xsd (this is added below Price) a. This is to support CAP requirements for residential lettings **Other Changes:** Added this appendix

API version	*Schema Change*
v8	**Schema Changes in this version:** 1. Encoding changed from 1.1. ISO-8859-1 (plus HTML escape codes for £, € and ² characters) to 1.2. UTF-8 (plus HTML escape codes for £, € and ² characters) **Other Changes:** 1. Added Encoding section 2. Added Appendix 4 3. Various updates to notes to the API calls (see Document History)
v9	**Schema changes in this version:** 1. New Lettingsfee element added to the propertyexport.xsd 2. New Newbuild element added to the propetyexport.xsd a. This will either return a value as per the propertyexport.xsd annotation or where we hold a NULL value, nil="true" 3. Expanded field length for **Bullets** element from 150 to 200 in the propetyexport.xsd in anticipation of a database structure change a. This will not immediately impact the length of the data received as database changes plus changes to the property upload from the Agent's application to vebra.com will need to be implemented first. **Other changes:** Expanded Note 1 to the Get Changed Properties call to take account of updates to the Lettings Fee data

API version	*Schema Change*
v10	**Schema changes in this version:** 1. Addition of new **rmqualifier value 16 – Coming Soon** – in the rightmovetypes.xsd a. This is returned as <rm_qualifier> in the Get Property call 2. Addition of new **software** field within the **reference element** in the propertyexport.xsd 3. Removed "optional" from **display attribute on price element** as it is always present 4. Addition of new values that are unique for Lettings properties (residential and commercial) within the **propstatus** element in the propertyrelatedtypes.xsd: a. Please refer to new **Appendix 5 – Lettings Statuses** for further details 5. The existing **file type 6** referenced in the propertyrelatedtypes.xsd as **Not used** has been updated to **Full details** a. This will include the brochure link for property of vebra.com in the Get Property call **b.** It is not included in the Get Changed Files call **Schema documentation changes:** 1. **Exportapi.xsd** - documentation updated re a. FirmID b. BranchID c. branches 2. **Propertyexport.xsd** - documentation updated re a. price node now includes documentation for display attribute b. propertyType node updated documentation re area element **c.** propertyType node updated documentation re landarea element **d.** propertyType node updated documentation re rm_type element **3. Propertyrelatedtypes.xsd** **a.** Updated documentation re fileTypes where Type = 6 from **Not used** to **Full details**

API version	*Schema Change*
	Other changes: 1. Updated Note 2 to the Get Property call regarding EPC graphs included in the files element where the property EPC values are non-zero and we do not hold associated EPC files for the property 2. Added new Appendix 5 – Lettings Statuses 3. Update to Appendix 1 – EPC Image Files 4. Update to Appendix 3 – Schema and Main Changes History **5.** Updated wording in **Summary of API Functionality** section re supported versions. **6.** Removed wording in **Get Property List** section relating to beta version
v11	**Schema changes in this version:** **Propertyexport.xsd** **1. All date/time related fields have been changed to xs:DateTime** using the ISO date format **yyyy-MM-ddThh:mm:ss.sss**, all times are local GMT/BST, this affects the following elements; a) available b) uploaded c) solddate d) leaseend e) instructed f) updated g) letdate (new) 2. Addition of new **adverts** element 3. Addition of new **letdate** element 4. Addition of support for **Per Person Per Week** (pppw) rentals in the price node 5. Addition of new **tenure** element 6. Addition of new **spaces** attribute added to the **parking** element 7. Addition of new **userfeatures** element 8. Changed field length for **rooms** element from 150 to 120 9. Updated the **EnergyRating constraint** to allow values in the range of 1-200 a) Applies to the **energy_efficiency** and **environmental_impact** elements

API version	*Schema Change*
v11	**Schema documentation changes:** 10. Documentation included or updated for all above fields in the Propertyexport.xsd 11. Annotation added for datetime formats – ISO format yyyy-MM-ddThh:mm:ss.sss a) Previous versions did not include milliseconds **Other Changes:** 1. Rework to support use of exclusions from the export and removal or addition of the exclusion. This applies to all versions of the API b) **Appendix 2 – Properties Excluded from the export** re-written to reflect new behaviour 2. All URLs for files that are served by Property Software Group products now served as **HTTPS** for the following domains a) images.portalimages.com b) `alto-live.s3.amazonaws.com` c) `webcdn1.jupix.co.uk` d) vebra.com 3. Update to Appendix 3 – Schema and Main Changes History 4. Other changes to this user guide from previous versions – for information for those upgrading a. Removed Note 1 from API Call Lists -> Get Property List b. Removed Note 3 from API Call Lists -> Get Property c. Removed Note 3 from API Call Lists -> Get Changed Files

v12	**Schema changes in this version:** **Propertyexport.xsd** **The following elements have been removed from this version** 4. premium 5. service_charge 6. rateable_value 7. leaseend 8. groundrent **The following elements have been added to replace and expand on the above elements (except premium).** council_tax with the child elements • band • reason (reason why the property is TBC or Exempt) domestic_rates – for Northern Ireland properties service_charge_amount with the attributes • frequency • review_period ground_rent_amount with the attributes • percentage_increase • review_period leasehold with the child elements • expiry (was leaseend) • years_remaining • shared_ownership • rent amount frequency • percentage

Appendix 4 – HTML Escape Code Support

The following table details the elements and attributes for which the HTML escape codes are returned in the Get Property call for:

1. Pound Symbol (£)
    a. Escape code = £
2. Euro Symbol (€)
    a. Escape code = €
3. Superscript to denote Square Feet / Metres (²)
    a. Escape code = ²

Element	***Attribute***
<address>	<name> <street> <locality> <town> <county> <custom_location> <display>
<price>	<qualifier>
<comm_rent>	
<rentalfees>	
<type>	
<type>	
<userfield1>	
<description>	
<lettingsfee>	
<paragraphs>	<name> <metric> <imperial> <mixed> <text>
<bullets>	<bullet id “*n*’’>
<files>	<name>

Appendix 5 – Lettings Statuses

The following is an explanatory note re property statuses for Lettings properties – both Residential and Commercial.

This relates to the values in the <propstatus> element in the propertyrelatedtypes.xsd and values that will be returned in the <web_status> element in the results for the **Get Property** call.

To remove any ambiguity with property statuses across Sales and Lettings, the Property Software Group has introduced a new set of property statuses that are unique for Lettings properties: both Residential and Commercial. properties and Commercial properties (both Lettings and Sales).

The New Lettings Statuses 100, 101, 102, 103 and 104 are included in the Propertyrelatedtypes.xsd within the propstatus definitions.

The existing values of 0, 1, 2, 3 and 4 will continue to be used for Residential and Commercial Sales properties.

The Client Feed Export API v10+ supports both the Old Shared Statuses and the New Lettings Statuses.

The values Old Shared Statuses are:

Old Shared Statuses		
Status	***Residential Lettings Definition***	***Residential Sales / Commercial Definition***
0	To Let	For Sale
1	Let	Under Offer
2	Under Offer	Sold
3	Reserved	Sold Subject to Contract (SSTC)
4	Let Agreed	For Sale By Auction

The New Lettings Statuses are:

New Lettings Statuses	
Status	***Lettings Definition***
100	To Let
101	Let
102	Under Offer
103	Reserved
104	Let Agreed

The software products are being upgraded to use the New Lettings Statuses, but there will be a transitional period whilst this takes place.

Where the software product has not been updated

Old Lettings Statuses 0-4 inclusive will still be used for Residential Sales properties and Commercial properties (both Sales and Lettings).

Hence calls to the Client Feed Export Get Property call will return results with <web_status> = the old Lettings Statuses 0-4 where the property meets the following criteria

1. Database = 2
2. Database = 5 or 118 and the <commercial> element has <transaction> = rental

Where the software product has been updated

New Lettings Statuses 100-104 inclusive will be used for new lettings properties or pre-existing ones that are re-uploaded: both Residential and Commercial.

Properties previously uploaded and still 'current' that have not been re-uploaded since the software product was updated will still have the old Lettings Statuses 0-4.

Hence calls to the Client Feed Export Get Property will return results with <web_status> that could be a mix of the old Lettings Statuses 0-4 and new Lettings Statuses 100-104 e.g.

1. A Residential letting property with Database = 2 that is To Let may have a web_status = 0 or 100
2. A Commercial letting property with Database = 5 or 118 that is To and the <commercial> element has <transaction> = rental may have a web_status = 0 or 100