
Web Service Description

Product: InfoCasa Client Web
Service

Change record

Date	Author	Version	Change Reference
30 Nov 2005	Thomas Bruusgaard	0.1	Initial draft
30 May 2006	Miguel Ángel Gómez	0.2	Added 'MY' search value
12 Jul 2006	Miguel Ángel Gómez	0.3	Added GetMap operation
15 Jan 2007	Miguel Ángel Gómez	0.4	Added 'LMD' search value and 'LNK' property field
20 Feb 2007	Miguel Ángel Gómez	0.5	Added 'DR', 'KN', 'LR', 'NR', 'NS', 'NF', 'BY', 'ST', 'SPR', 'MO', 'CD', 'LMD', 'SHD', 'TN', 'FU', 'KI', 'OR'
12 Jun 2007	Miguel Ángel Gómez	0.6	Added 'RK', 'CD' and 'LMD' to sort fields. 'LA' and 'STA' to additional fields. Images subtypes and doc type to attachments. RN to CountProperties and SearchProperties
26 Sep 2007	Miguel Ángel Gómez	0.7	Added 'IDRE' to search values. 'TU', 'UTM', 'MAPX', 'MAPY' to additional fields. Search params applied when WH is sent.
26 June 2008	Miguel Ángel Gómez	0.8	Added 'TS' to GetProperties. Added 'C', 'ND', 'WH' to additional fields. Added new supported values for ci.
10 June 2009	Miguel Ángel Gómez	0.9	Added short term rental operations. Added 'RTM', 'CID', 'COD', 'STA' as search value. Added 'MRP', 'RPF', 'RTM', 'SLP', 'RPP', 'RPM', 'RPL' to additional fields. Added 'RN' as sort field.
13 Sept 2010	Miguel Ángel Gómez	1.0	Added 'TAG' search value. Sort fields by all rental prices. Appendix F with rental price frequency

			values. Added WebUser operations. Added 'HTMLT', 'HTMLD', 'HTMLK', 'FURL', 'DER' to additional fields.
28 Mar 2011	Miguel Ángel Gómez	1.1	Added GetAgencies operation.
20 Jun 2012	Miguel Ángel Gómez	1.2	Added rtg input param to GetProperties and GetPropertyDetails; desc input param to GetProperties; field type TAG, to GetProperties and GetPropertyDetails response.
06 Aug 2012	Miguel Ángel Gómez	1.3	Allow to search properties with 0 bedrooms and/or 0 bathrooms.
05 Mar 2013	Miguel Ángel Gómez	1.4	Added lmd field to attachments. Increased margin days in rental availability.
10 Apr 2013	Michael Cooling	1.5	Added GetReport operation. Added new order fields: 'PS' and 'BS'. Added 'LTAX', 'BTAX', 'CFEE', 'DERDEP', 'DERPAY' to additional fields. Added 'RPF' to search values.
09 Dec 2014	Miguel Ángel Gómez	1.6	Added 'MAD' to order fields. Added support for multiple order fields. Added GetEnergyPerformanceImage operation. Added 'EER', 'EIR', 'ZIP' to additional fields. Added support to return documents in GetAttachment operation.
02 Aug 2017	Miguel Ángel Gómez	1.7	Added 'rln' input param to GetPropertyDetails. Added 'BSX', 'PSX', 'TSX' to search values.

			Added 'fbdesc' input param to GetProperties and GetPropertyDetails. Added 'TRID' to additional fields. Added 'links' to GetPropertyDetails response.
20 Oct 2017	Miguel Ángel Gómez	1.8	Added 'RAFE', 'ROEX', 'PVAL', 'PSPE' to additional fields in GetPropertyDetails
09 Aug 2019	Miguel Ángel Gómez	1.9	Added Appendix H – Getting properties images
30 Jul 2020	Miguel Ángel Gómez	1.10	Deprecated old operations: GetMap, GetSavedSearchInfo, GetSavedSearch, RegisterWebUser, ActivateWebUser, LogInWebUser, ModifyWebUser, ModifyWebUserPwd, RememberWebUserPwd Added 'AddPropertyEnquiry' operation.
26 Feb 2021	Miguel Ángel Gómez	1.11	Added 'DECPX' to additional fields in GetPropertyDetails.
22 Sep 2021	Miguel Ángel Gómez	1.12	GetAttachment, GetMap, GetReport and GetEnergyPerformanceImage have been discontinued

Table of contents

Change record.....	2
Table of contents	1
Introduction	4
Overview	5
Licensing	7
Common xml elements.....	8
CountProperties.....	9
SearchProperties	10
Search parameters.....	11
Search Values	12
Area Group (AG)	14
Web Highlight (WH)	14
Number of floors, receptions or storeys (NF, NR, NS)	14
Min. and Max. built size, plot size and terrace size (BS, BSX, PS, PSX, TS, TSX)	14
Built year (BY)	14
Creation date (CD)	15
Last Modification date (LMD)	15
Multiple listing flag (MLS)	15
New development flag (ND)	15
Reference number (RN)	16
Country (C)	16
Has Price and Has Image (HP & HI)	16
My listings flag (MY)	17
Listing agency (IDRE)	17
Registered office (AREG)	17
Rental term (RTM)	17
Check-in date, Check-out date (CID, COD)	18
Status (STA)	18
Favourite group name (TAG)	18
Address (ADR)	18
Rental Price Frequency (RPF)	19
Sort	19

Property details	21
Additional fields	21
Attachments	25
Links	26
GetProperties	27
GetPropertyDetails	29
GetPropertyAvailability	31
Output	31
Bookings	32
GetRentalCalendar	33
Output	33
Periods	34
GetAttachment [Deprecated]	35
Output	36
GetMap [Deprecated]	38
GetReport [Deprecated]	39
GetEnergyPerformancelmage [Deprecated]	40
Output	40
GetKeys	41
Key types	41
Output	42
Example 1 – Property types	42
Example 2 – Currency	43
GetAreas	44
Search criteria (sc)	45
Output	45
GetAreaGroups	47
Output	47
GetSavedSearchInfo [Deprecated]	49
Output	49
GetSavedSearch [Deprecated]	50
GetAgencies	50

Output	51
WebUser [Deprecated]	51
RegisterWebUser [Deprecated]	52
Output	53
ActivateWebUser [Deprecated].....	54
Output	54
LoginWebUser [Deprecated].....	54
Output	55
ModifyWebUser [Deprecated].....	55
Output	55
ModifyWebUserPwd [Deprecated].....	56
Output	56
RememberWebUserPwd [Deprecated].....	56
Output	57
AddPropertyEnquiry	57
Output	60
Example work flows.....	61
Building a search page	61
Initialising a search	61
Pagination	62
Quick links	63
Property details	64
Appendix A – Country identifiers	65
Appendix B – Key types	66
Appendix C – Area levels	66
Appendix D – Predefined Area Groups	66
Appendix E – Currency codes	66
Appendix F – Rental Price Frequency values.....	67
Appendix G – Property status values	67
Appendix H – Getting properties images.....	68

Introduction

This document describes the Client Web Service provided by InfoCasa. It is meant to be a guideline for someone who is interested in implementing a consumer of the web service.

The readers of this document are assumed to have good knowledge of Xml Web Services and WSDL. It is not the purpose of this document to describe in details the WSDL document for the InfoCasa Client Web Service.

Note: InfoCasa CRM, S.L. does provide a Web Service Consumer as part of a client web kit (InfoCasa Web SDK).

Overview

The Client Web Service (which can be found at <http://api.infocasa.com/clientservice.asmx>) enables a licensed client of InfoCasa to integrate a search of its properties into a web site.

All information returned from the web service is publicly available.

The web service consists of the following functions:

Function	Description
CountProperties	Returns the number of properties which match a certain criteria.
GetAreaGroups *	Returns a list of area groups, like “Costa del Sol” or “Fungirola to Málaga”.
GetAreas *	Returns a list of areas where agent has properties.
GetAttachment (obsolete)	Returns an image (using the DIME protocol)
GetKeys *	Returns a list of “id – description” pairs. E.g. property types and currency.
GetProperties	Returns a list of between 0 – 20 properties. Used for quick access when paging.
GetPropertyDetails	Returns public details about a specific property.
GetPropertyAvailability	Returns whether a property is available for rent and a list of rental bookings of a property given a date range. It can also return rental calendar details as in the GetRentalCalendar operation.
GetRentalCalendar	Returns calendar details and a list of season date ranges given a date range and either a calendar id or a property id.
GetSavedSearch **	Returns a search which has been saved in the InfoCasa system for a specific client. This search can then automatically be passed to the CountProperties or SearchProperties functions described here.
GetSavedSearchInfo **	Returns a list of “id – description” pair

	of saved searches in the InfoCasa system.
SearchProperties	Returns a list of 0 – 200 property id's depending on the search criteria passed.

* These functions aid the client to populate search criteria on a web page.

** These functions aid the client to add quick link searches to a web site.

Licensing

The web service is only available to licenced users. Each function in the service takes a Soap Header of the following format:

```
<soap:Header>
  <licenceHeader xmlns="urn:schemas-infocasa-com:client-
service:2005-11">
    <lk>string</lk>
  </licenceHeader>
</soap:Header>
```

The <lk /> string must be replaced by a licence key supplied by InfoCasa.

Common xml elements

In every operation there are a few common xml elements which repeat. This section describes these elements.

Element	Description
idre	A five character long identifier of an InfoCasa client.
ci	Culture info. This parameter specifies what language the results will be brought back in. Currently supported values are: • en - for english • es - for spanish • de - for german • nl - for dutch • da - for danish • fr - for french • sv - for swedish • no - for norwegian • fi - for finnish • ca - for catalan • it - for italian • ru - for russian • el - for greek • cs - for czech • sk - for slovak • pl - for polish Defaults to en for English.

Note: These elements are not described in the sections below about the specific operations.

CountProperties

The count properties input parameters are described in the following section below [Search parameters](#).

The output of this operation is simply the number of properties matching the Search parameters passed:

```
<countProperties xmlns="CountProperties_out">
  <np>int</np>
</countProperties>
```

Element	Description
np	Number of properties.

SearchProperties

The SearchProperties input parameters are described in the following section below [Search parameters](#).

The output of this operation is an array of property identifiers. The number of identifiers returned is between 0 and 200:

```
<propertyIds version="string" xmlns="SearchProperties_out">
  <id>int</id>
  <id>int</id>
</propertyIds>
```

Element	Description
id	Property identifier. This identifier is used as an input to the GetProperties and GetPropertyDetails operations.

Search parameters

Both the CountProperties and SearchProperties web service functions take the same XML as input.

```
<searchProperties xmlns="SearchProperties_in">
  <idre>string</idre>
  <ci>string</ci>
  <sf>boolean</sf>
  <minbd>int</minbd>
  <maxbd>int</maxbd>
  <minba>int</minba>
  <maxba>int</maxba>
  <minp>decimal</minp>
  <maxp>decimal</maxp>
  <propType>int</propType>
  <areas>
    <id>int</id>
    <id>int</id>
  </areas>
  <keys>
    <id>int</id>
    <id>int</id>
  </keys>
  <searchValues>
    <sv ty="string">
      <val>string</val>
    </sv>
    <sv ty="string">
      <val>string</val>
    </sv>
  </searchValues>
  <sort>
    <fld>string</fld>
    <desc>boolean</desc>
  </sort>
  <index>int</index>
</searchProperties>
```

Field	Description
sf	Sale flag. Set to true to search for properties for sale
minbd / maxbd	Minimum and maximum number of bedrooms. Default value, '0', means no condition on that field, to search for 0 bedrooms, a '-1' value must be set.
minba / maxba	Minimum and maximum number of bathrooms. Default value, '0', means

	no condition on that field, to search for 0 bathrooms, a '-1' value must be set.
minp / maxp	Minumum and maximum price.
propType *	Property type.
areas	Array of area id's. These id's correspond to the id's of areas returned from the GetAreas function.
keys	Array of key id's. These id's correspond to the id's of keys returned from the GetKeys function.
searchValues	Collection of additional search values (<sv /> elements). See the Search Value section later on in this document.
sort	Sort order of results returned. This is ignored for the CountProperties function.
index	Index of the 200 properties group to return. Default value is 0.

* DEPRECATED, please use keys instead. The propType value is actually a key value, but has a short cut in the search parameters as it is a commonly searched field. To search for more than one property types, simply add the property type id's to the array of keys.

Search Values

The <sv /> element contain additional search criteria or filter values. It has the following sub elements:

Field	Description
ty	Type attribute. Type of search field. The following values are valid • HI – Has images * • HP – Has price * • C – Country * • AG – Area group • WH – Web highlight • NR – Number of receptions • NS – Number of storeys • BS – Min. built size • BSX – Max. built size • PS – Min. plot size • PSX – Max. plot size

	• TS – Min. terrace size • TSX – Max. terrace size • NF – Number of floors • BY – Build year • RN – Reference number • CD – Creation date • LMD – Last Modification Date • MLS – MLS listing flag • ND – New development flag • MY – My listings flag • IDRE – Listing agency • AREG – Registered office • RTM – Rental term (1= Short term, 2 = Long term) • CID – Check-in date (when searching for rentals) • COD – Check-out date (when searching for rentals) • STA – Property Status • TAG – Favourite group name • ADR – Property address • RPF – Rental Price Frequency. See Appendix F – Rental Price Frequency values See the sections below for a detailed description of currently available values.
val	The value of the search field. The format of this value depends on what type.
op	The operator on the search value, currently the supported values are: • = – Equal (default) • != – Not equal This field is optional and ignored for all search values except for IDRE, AREG, TAG and ND.

* The **Has Price**, **Has Images** and **Country** fields are used as filter values, to filter what comes back from the InfoCasa system. The <filters /> xml element can be seen in other web service operations also, such as GetAreas, GetKeys and GetAreaGroups.

Area Group (AG)

The area group search value can be set to the identifier of an Area Group returned from the GetAreaGroups web service operation. The value must be an integer, as this is the type of the Area Group identifier.

Example Xml:

```
<sv ty="AG">
  <val>1</val>
</sv>
```

Web Highlight (WH)

The web highlight search value, when added to the search parameters will bring back all properties which have been marked as Web Highlight in the InfoCasa system. This search value will be combined with all the other criteria specified. Previous to 01/10/2007, all other criteria were ignored. This value should be set to "true" to search for highlighted properties.

```
<sv ty="WH">
  <val>true</val>
</sv>
```

Number of floors, receptions or storeys (NF, NR, NS)

These values are integer values, and should be set in a similar manner.

```
<sv ty="NF">
  <val>3</val>
</sv>
```

Min. and Max. built size, plot size and terrace size (BS, BSX, PS, PSX, TS, TSX)

These values are integer values, and should be set in a similar manner. The unit of measurements are square meters (m²).

```
<sv ty="BS">
  <val>250</val>
</sv>
```

Built year (BY)

The built year is a four digit integer value and should be set in a similar manner.

```
<sv ty="BY">
```

```
<val>1994</val>
</sv>
```

Creation date (CD)

The creation date is of type DateTime with the format YYYY-MM-DD HH:mm:ss, but could contain a date only (YYYY-MM-DD). It will filter properties with a creation date greater than the one specified.

```
<sv ty="CD">
  <val>2005-11-28</val>
</sv>
```

Or:

```
<sv ty="CD">
  <val>2007-02-20 13:00</val>
</sv>
```

Last Modification date (LMD)

The last modification date is of type DateTime with the format YYYY-MM-DD HH:mm:ss, but could contain a date only (YYYY-MM-DD). It will filter properties with a last modification date greater than the one specified.

```
<sv ty="LMD">
  <val>2007-01-15</val>
</sv>
```

Or:

```
<sv ty="LMD">
  <val>2007-02-20 13:00</val>
</sv>
```

Multiple listing flag (MLS)

This search value is of type integer and the allowed values are 0 and 1. The value should be set in a similar manner.

```
<sv ty="MLS">
  <val>1</val>
</sv>
```

New development flag (ND)

This search value is of type integer and the allowed values are 0, 1, 2 or 3; 0 for resale properties, 1 for properties under construction, 2 for properties in project phase and 3 for new built / development properties. The <op> element is taken into account for this search value so searches can also be done with the != operator. The value should be set in a similar manner.

```
<sv ty="ND">
  <val>1</val>
</sv>
```

Or:

```
<sv ty="ND">
  <val>0</val>
  <op>!=</op>
</sv>
```

In the previous example properties that are not resale would be returned in the results, that is, it will return properties set as “Under construction”, “Projected construction” or “New development”.

Reference number (RN)

Setting the reference number search value will search for properties which reference contains this reference. The reference number is of type string, and should be set in the following manner:

```
<sv ty="RN">
  <val>ref 001</val>
</sv>
```

Country (C)

The country search value is available so that properties can be filtered by country by default. The value is of type integer and takes a country identifier. See [Appendix A – Country identifiers](#) for a list of currently used identifiers in the InfoCasa system. To filter searches for a specific country, the search value should be specified in the following manner (3 = Spain):

```
<sv ty="C">
  <val>3</val>
</sv>
```

Note: The country search value is ignored once any area id's are added to the search, as these will take precedence.

Has Price and Has Image (HP & HI)

These search values have two possible values, true or false. They are used primarily to filter out properties which have not had a price, nor image specified. The search value should be specified like this:

```
<sv ty="HP">
  <val>true</val>
</sv>
```

My listings flag (MY)

This search value is of type integer and the allowed values are 1, 2 or 3. 1 filters by properties listed by the agency, 2 by properties listed by any agency in the same group and 3 by properties not listed by any agency in the group.

```
<sv ty="MY">
  <val>1</val>
</sv>
```

Listing agency (IDRE)

This search value can be used to search for properties listed or not listed by a given agency code. It is a string of 3 or 5 characters length.

```
<sv ty="IDRE">
  <val>ABX</val>
</sv>
```

Or:

```
<sv ty="IDRE">
  <val>ABXAI</val>
  <op>!=</op>
</sv>
```

In the previous example properties that are not listed by that agency would be returned in the results.

Registered office (AREG)

This search value can be used to search for properties registered or not registered by a given agency code. It is a string of 3 or 5 characters length. With this search value the op field is not ignored.

```
<sv ty="AREG">
  <val>ABX</val>
</sv>
```

Or:

```
<sv ty="AREG">
  <val>ABXAI</val>
  <op>!=</op>
</sv>
```

Rental term (RTM)

This search value can be used to search for properties that are for rent for a specific term (2=long or 1=short).

```
<sv ty="RTM">
  <val>1</val>
```

```
</sv>
```

Check-in date, Check-out date (CID, COD)

This search value can be used to search for properties for rent that are available between these to dates.

```
<sv ty="CID">
  <val>2007-02-20 13:00</val>
</sv>
```

Status (STA)

This search value can be used to search properties by status: reserved, archived, available, etc.

See [Appendix G – Property status values](#) for a list of currently used property status in the InfoCasa system.

```
<sv ty="STA">
  <val>int</val>
</sv>
```

Favourite group name (TAG)

This search value can be used to search for properties grouped or not grouped in a favourite group in Infocasa application.

```
<sv ty="TAG">
  <val>my-featured-properties</val>
</sv>
```

Or:

```
<sv ty="TAG">
  <val>my-featured-properties</val>
  <op>!=</op>
</sv>
```

Address (ADR)

This search value can be used to search for properties with addresses that contain all of the words in this search value.

```
<sv ty="ADR">
  <val>elviria marbella</val>
</sv>
```

Rental Price Frequency (RPF)

When searching for rentals with a price range (minp and/or maxp) this value is very useful because it determines if the price specified is per day, week, month, etc. See [Appendix F – Rental Price Frequency values](#).

If this value is not specified and RTM (rental term) is also not specified, a monthly price is assumed, if short term is specified in RTM, weekly is assumed and if long term is specified in RTM, monthly is assumed.

```
<sv ty="RPF">
  <val>1454</val>
</sv>
```

Sort

The <sort /> element specifies the order by which results are returned. It has the following sub elements.

Field	Description
fld	Comma separated list of fields, including asc / desc modifier, i.e. PR desc, RN asc. The following values are valid • PR – Price • LO – Location (area) • BR – Bedrooms • RK – Property Ranking • CD – Creation Date • LMD – Last Modification Date • MAD – Made Available Date • RN – Reference Number • RPL – Rental Price Low • RPM – Rental Price Medium • RPP – Rental Price Peak • RPS – Smallest rental price out of all the rental prices. • BS – Built size. • PS – Plot size. Defaults to PR (Price)
desc	Descending flag. Set to true for a descending sort, false for an ascending sort. Obsolete, use 'asc' or 'desc'

	modifier next to the field name.
--	---

Property details

The details of a property are returned from GetProperties and GetPropertyDetails operations. The difference between the two operations is that the GetProperties returns a list of details (details of between 0 and 20 details), where the description is limited to 200 characters.

The output XML looks like this:

```
<propertyDetails version="string" xmlns="Properties_out">xml
  <properties>
    <p>xml
      <pr>decimal</pr>
      <bd>int</bd>
      <pt id="int">string</pt>
      <ar id="int">string</ar>
      <fields xsi:nil="true" />
      <rn>string</rn>
      <attachments xsi:nil="true" />
      <links xsi:nil="true" />
    </p>
    <p ... />
  </properties>
</propertyDetails>
```

Field	Description
pr	Price of property.
bd	Number of bedrooms
pt	Property type.
ar	Area where property is located
fields	See Additional fields
rn	Property reference number.
attachments	See Attachments
links	List of external links added to the property. Links are only returned if input param 'rln' = true is received.

Additional fields

Contain additional fields of a property.

The <fld /> xml looks like this:

```
<fld tp="string" dt="string">
  <val>string</val>
  <txt>string</txt>
  <ci>string</ci>
```

</fld>

Field	Description
tp	Attribute. Type of field. Following values are available at the moment: • ER – Exchange Rate (Additional Key value) • BS – built size • PS – plot size • TS – terrace size • BR – number of bathrooms • POA – Price on application • FE – Feature • VI – Views. Only in GetPropertyDetails operation. • PO – Pool • GA – Garden • PA – Parking • PI – Property identifier (internal) • DE – description (limited to 150 characters in the GetProperties operation) • ADA – Area description (ascending) • ADD – Area description (descending) • PRO – Province identifier and name (see Area output) • C – Country identifier (see Area output) • RP – Rental Price HIGH for the RPF (See RPF field below). • LNK – External link to Virtual Tour, brochure, etc. • DR – Number of dinning rooms • KN – Number of kitchens • LR – Number of living rooms • NR – Number of receptions • NS – Number of storeys (levels)

	• NF – Floor number • BY – Built year • ST – Sales terms (a = Exclusive, b = Non exclusive) • SPR – Input selling price • MO – Input currency (see Currency codes) • CD – Creation date • LMD – Last modification date • SHD – Short description (accessibility field in Infocasa) • TN – Number of toilets • FU – Furnished feature • KI – Kitchen feature • OR – Facing feature • LA – Living Area • STA – Property Status • TU – Location type • UTM – GPS UTM Zone • MAPX – UTM Easting value • MAPY – UTM Northing value • ND – New Development (0 = Resale, 1 = Under const., 2 = In Project, 3 = New built / develop.) • WH – Web highlight (boolean) • FSL – For sale (1 = for sale) • MRP – Minimum rental period • RPF – Rental price frequency (see Appendix F – Rental Price Frequency values) • RTM – Rental term (1 = Short term, 2 = Long term) • SLP – Number of people the property sleeps (Only returned if the property is for short term rental) • RPP – Rental price PEAK for the RPF • RPM – Rental price MEDIUM for the RPF • RPL – Rental price LOW for the RPF
--	--

	• CALID – Rental calendar id. • RAFE – Rental admin fees • ROEX – Rental other monthly expenses. • TRID – Tourist Registry id. • HTMLT – HTML Title • HTMLD – HTML Description • HTMLK – HTML Keywords • FURL – Friendly URL. • DER – Rental description. • IDRE – Listing agency. • AREG – Registered office. • TAG – Tag name. Can appear multiple times. • LTAX – Local Tax (IBI). Only in GetPropertyDetails operation. • BTAX – “Basura” (Rubbish) Tax. Only in GetPropertyDetails operation. • CFEE – Community Fee. Only in GetPropertyDetails operation. • DERDEP – rental description deposits. Only in GetPropertyDetails operation. • DERPAY – rental description payment method. Only in GetPropertyDetails operation. • EER – Energy Efficiency Rating (Energy consumption). • EIR – Environmental Impact Rating (CO2 Emissions). • ZIP – Postal code of the area assigned to the property. • PVAL – Id of the valuer assigned to the property. • PSPE – Id of the salesperson assigned to the property. • DECPX – building, complex or promotion description. Only in GetPropertyDetails operation.
dt	Attribute. Data type.

	Following values are available at the moment: • decimal • int • boolean • string
val	Value of the field. (Identifier in case of a key value)
txt	Text value of a key field.
ci	Culture info. This parameter specifies the language of the field. Currently it is only returned in FURL field in GetPropertyDetails operation.

Attachments

An agent can attach files to a property (normal pictures, location maps, floorplans, PDF documents, 360° panoramic images), those attachments set as visible in web will be returned by the GetPropertyDetails operation.

The attachments have the following format:

```
<attachments>
  <at tp="string" st="string">
    <ca>string</ca>
    <na>string</na>
    <lmd>datetime</lmd>
  </at>
  <at ... />
</attachments>
```

Field	Description
tp	Attribute. Type of attachment. Following values are available at the moment: • image – image related to property • doc – document attached to property
st	Attribute. Subtype of attachment. Following values are available for type image: • blank – Normal image (picture).

	• MAP – Location Map or indications image of property. • 360 – 360° panoramic image of property. Need Java Applet, Flash or JavaScript viewer to be shown on web. • FLP – FloorPlan image of property. Following values are available for type doc: • blank – No subtypes exists for documents.
ca	Caption / description of image in the specified language.
na	Name of image, needed to call to the GetAttachment operation to bring back the image.
lmd	Last modification date/time of the attachment in the format: YYYY-MM-DDTHH:mm:ss.

Links

An agent can add multiple external links to a property, currently only videos can be added through InfoCasa web application. Links are only returned by GetPropertyDetails operation if “rln” input parameter is received with a “True” value.

The links have the following format:

```
<links>
  <link tp="string" cat="string">
    <url>string</url>
    <ca>string</ca>
  </link>
  <link ... />
</links>
```

Field	Description
tp	Attribute. Type of link. Following values are available at the moment: • video – The URL contain a

	video, i.e. an MPEG, YouTube or Vimeo video.
cat	Attribute. Category of the link. Following values are available for "video" type: • main – Main video. • additional – Additional video. • area – Video of the area.
url	URL of the link.
ca	Caption / description of the link.

GetProperties

The GetProperties operation returns a list of between 0 and 20 [property details](#).

The input of the operation is defined as the following:

```
<getProperties xmlns="GetProperties_in">
  <idre>string</idre>
  <ci>string</ci>
  <sf>boolean</sf>
  <arc>boolean</arc>
  <ids>
    <id>int</id>
    <id>int</id>
  </ids>
  <rtg>boolean</rtg>
  <descl>int</descl>
  <fbdesc>string</fbdesc>
  <sort>
    <fld xmlns="SearchProperties_in">string</fld>
    <desc xmlns="SearchProperties_in">boolean</desc>
  </sort>
</getProperties>
```

Field	Description
sf	Sale flag. Needed to be able to sort by the right price (sale or rental). Defaults to true.
arc	Archived. Set to true if need to search on archived properties. Defaults to false.
ids	Array of property identifiers. These identifiers are returned from the SearchProperties operation.

rtg	Request Tags (Optional). True if want to receive tags (property groups in Infocasa) properties are assigned to. Default value is false.
descl	Description Length (Optional). By default web service will return up to 200 characters of the property description and short description. Can be a positive number, to get a specific number of characters or -1 to get the full description.
fbdesc	Fallback description (Optional). Defines which description will be returned in case there were no description in the requested language. Possible values: 'auto' – Auto generated description. '{lang}' – Any valid lang id, i.e. 'en'. Default value is 'auto'.
sort	See the Sort section above.

GetPropertyDetails

The GetProperties operation returns 0 or 1 [property details](#) depending whether the property id or the reference number passed to the operation are valid.

The input to the operation is defined as such:

```
<getPropertyDetails xmlns="GetPropertyDetails_in">
  <idre>string</idre>
  <ci>string</ci>
  <pi>int</pi>
  <rn>string</rn>
  <arc>boolean</arc>
  <sf>boolean</sf>
  <rtg>boolean</rtg>
  <rln>boolean</rln>
  <fbdesc>string</fbdesc>
</getPropertyDetails>
```

Field	Description
pi	Property identifier. This identifier is returned from the SearchProperties operation.
rn	Property reference number. This reference number is passed back as part of the Property details . The reference number would normally be used when a client is searching for a property using a specific reference number.
arc	Archived. Set to true if need to get details of an archived property. Defaults to false.
sf	SaleFlag. Set to true to only return the property details if the property is for sale, if set to false the property details will only be returned if the property is for rent. If the field is not sent then it is ignored and the details will always be returned.
rtg	Request Tags (Optional). True if want to receive tags (property groups in Infocasa) property is assigned to.

	Default value is false.
rln	Request Links (Optional). True if want to receive property links. Default value is false.
fbdesc	Fallback description (Optional). Defines which description will be returned in case there were no description in the requested language. Possible values: 'auto' – Auto generated description. '{lang}' – Any valid lang id, i.e. 'en'. Default value is 'auto'.

GetPropertyAvailability

The GetPropertyAvailability operation returns 0 or 1 [availability details](#) depending whether the property id is valid or the property is defined as a short term rental.

The input to the operation is defined as such:

```
<getPropertyAvailability xmlns="GetPropertyAvailability_in">
  <idre>string</idre>
  <ci>string</ci>
  <pi>int</an>
  <cid>date</cid>
  <cod>date</cod>
  <days>int</days>
  <calendar>boolean</calendar>
</getPropertyAvailability>
```

Field	Description
cid	Rental check-in date.
cod	Rental check-out date. Maximum allowed date range, that is, [cod-cid] is 180 days.
days	Margin of days before cid and after cod to get the bookings for. Maximum allowed value is 200.
calendar	Indicates whether we want the details of rental calendar associated to the property.

Output

The output of the operation looks like this:

```
<PropertyAvailability xmlns="Properties_out">
  <available>boolean</available>
  <bookings/>
  <RentalCalendar xmlns="RentalCalendar_out"/>
</PropertyAvailability>
```

Field	Description
available	Boolean value indicating if the property is available between the cid

	and cod dates.
bookings	Array of booking date ranges the property is booked for (occupied). See Bookings .
RentalCalendar	Contains general information on the calendar and an array of season periods of the calendar for the date range we have enquired on. See GetRentalCalendar .

Bookings

Contain rental booking records of a property.

The <bookings /> xml looks like this:

```
<bookings>
  <bk cid="2008-12-24T00:00:00" cod="2008-12-28T00:00:00" ty="1327" />
  <bk cid="2009-02-04T00:00:00" cod="2009-02-06T00:00:00" ty="1327" />
  <bk cid="2009-02-13T00:00:00" cod="2009-02-22T00:00:00" ty="1328" />
  <bk cid="2009-03-03T00:00:00" cod="2009-03-10T00:00:00" ty="1443" />
</bookings>
```

Field	Description
cid	Attribute. Check-in date.
cod	Attribute. Check-out date.
ty	Attribute. Type of booking record. Possible values are: • 1327 Rented • 1328 Owners disposition • 1443 Cleaning / Maintenance

GetRentalCalendar

The GetRentalCalendar operation returns the details of the rental calendar given by the calid input parameter or by the calendar associated to the property given by the pi (property identifier) input parameter.

The input to the operation is defined as such:

```
<getRentalCalendar xmlns="GetRentalCalendar_in">
  <idre>string</idre>
  <ci>string</ci>
  <calid>int</calid>
  <pi>int</pi>
  <inid>date</inid>
  <endd>date</endd>
</getRentalCalendar>
```

Field	Description
calid	Rental calendar id.
pi	Property id.
inid	Initial date of the range we require.
endd	End date of the range we require. Maximum allowed date range, that is, [endd-inid] is 366 days.

Output

The output of the operation looks like this:

```
<RentalCalendar xmlns="RentalCalendar_out">
  <calid>51314</calid>
  <cname>Test Calendar</cname>
  <cdesc>Calendar to test with</cdesc>
  <periods/>
</RentalCalendar>
```

Field	Description
calid	Rental calendar id.
cname	Rental calendar name.
cdesc	Rental calendar description.
periods	Contains a group of season period records <prd/>. See Periods .

Periods

Contain season date ranges for a rental calendar.

The <periods/> xml looks like this:

```
<periods>
  <prd pid="2009-01-01T00:00:00" ped="2009-01-05T00:00:00"
sid="1415" />
  <prd pid="2009-01-07T00:00:00" ped="2009-02-01T00:00:00"
sid="1412" />
  <prd pid="2009-02-02T00:00:00" ped="2009-02-08T00:00:00"
sid="1413" />
  <prd pid="2009-02-09T00:00:00" ped="2009-02-15T00:00:00"
sid="1415" />
  <prd pid="2009-02-15T00:00:00" ped="2009-02-22T00:00:00"
sid="1412" />
  <prd pid="2009-02-23T00:00:00" ped="2009-02-28T00:00:00"
sid="1415" />
  <prd pid="2009-03-01T00:00:00" ped="2009-04-05T00:00:00"
sid="1413" />
  <prd pid="2009-04-05T00:00:00" ped="2009-04-12T00:00:00"
sid="1415" />
  <prd pid="2009-04-13T00:00:00" ped="2009-05-14T00:00:00"
sid="1412" />
</periods>
```

Field	Description
pid	Attribute period initial date.
ped	Attribute period end date.
sid	Attribute season id. This indicates the id for the season the range belongs to, this being: • 1415 Peak season • 1414 High season • 1413 Medium season • 1412 Low season

GetAttachment [Deprecated]

This operation is discontinued since 01/12/2021, please refer to [Appendix H – Getting properties images](#) section to retrieve property images or use the more recent XML-HTTP API.

The GetAttachment operation returns details of the attachment and a DIME part (see <http://msdn.microsoft.com/msdnmag/issues/02/12/DIME/> for more information on the DIME specification) containing the specified attachment of a property. Currently only standard images, floor plans, panoramic, directions and standard documents are supported by the web service.

The input to the operation is defined as such:

```
<getAttachment xmlns="GetAttachment_in">
  <idre>string</idre>
  <ci>string</ci>
  <an>string</an>
  <as>string</as>
  <ah>int</as>
  <tp>string</tp>
</getAttachment>
```

Field	Description
an	Attachment name. This name is returned in the attachments section of GetProperties and GetPropertyDetails operations.
as	Attachment size (width). Only applied to attachments of type image. Following values are available at the moment: • small (150 px) • medium (300 px) • large (640 px) • xlarge (1024 px) • xxlarge (1300 px)
ah	Attachment height. Only applied to attachments of type image and only taken into account if the 'as' (Attachment size) field has no value. The minimum value for this field is

	150, and the maximum is 640.
tp	Type of attachment. Optional parameter to indicate if a document attachment (tp=doc) is requested.

Output

The output of the operation looks like this:

```
<attachment tp="string" st="string" xmlns="GetAttachment_out">
  <na>string</na>
  <ca>string</ca>
  <lmd>dateTime</lmd>
</attachment>
```

Field	Description
tp	Attribute. Type of attachment. Following values are available at the moment: • image – image related to property • doc – document attached to property
st	Attribute. Subtype of attachment. Following values are available for type image: • blank – Normal image (picture). • MAP – Location Map or indications image of property. • 360 – 360° panoramic image of property. Need Java Applet, Flash or JavaScript viewer to be shown on web. • FLP – FloorPlan image of property. Following values are available for type doc: • blank – No subtypes exists for documents.
na	Name of image.

ca	Caption / description of image in the specified language.
lmd	Last modification date/time of the attachment in the format: YYYY-MM-DDTHH:mm:ss.

GetMap [Deprecated]

This operation is discontinued since 01/12/2021.

The GetMap operation returns a DIME attachment containing a Flash map of the area specified.

The input of the operation is defined as the following:

```
<getMap xmlns="GetMap_in">
  <idre>string</idre>
  <ci>string</ci>
  <id>string</id>
</getMap>
```

Field	Description
Id	Map name. If 'MapSearch.swf' is specified, the map container is returned; in other case a valid map name must be specified. For example, '3.swf' corresponds to Spain map and '27.swf' to Málaga map. When the required map does not exist, an error is returned.

GetReport [Deprecated]

This operation is discontinued since 01/12/2021, please use the more recent XML-HTTP API.

The GetReport operation returns a DIME attachment containing a PDF report. The input of the operation is defined as the following:

```
<getReport xmlns="GetReport_in">
  <idre>string</idre>
  <ci>string</ci>
  <sf>boolean</sf>
  <reportName>string</reportName>
  <currency>int</currency>
  <rt>string</rt>
  <ids>
    <id>int</id>
    <id>int</id>
  </ids>
</getReport>
```

Field	Description
sf	Sale flag. Set to true to generate report in sale mode, false to generate it in rental mode.
reportName	Name of the source report to generate. Mandatory.
currency	Currency code to be used in the report. (see Currency codes)
rt	Report type, optional, (PR for normal property reports, PH for photo based reports). If the value is not sent, PR will be assumed.
ids	Array of property ids the report will be generated for. Mandatory. 10 property ids maximum.

GetEnergyPerformanceImage [Deprecated]

This operation is discontinued since 01/12/2021, please use the more recent XML-HTTP API.

The GetEnergyPerformanceImage operation returns energy consumption and CO2 emissions values as well as a DIME attachment containing the energy performance image of a property.

The input of the operation is defined as the following:

```
<getEnergyPerformanceImageIn
xmlns="GetEnergyPerformanceImage_in">
  <idre>string</idre>
  <ci>string</ci>
  <pi>int</pi>
</getEnergyPerformanceImageIn>
```

Field	Description
pi	Property identifier. This identifier is returned from the SearchProperties operation.

Output

The output of the operation looks like this:

```
<GetEnergyPerformanceImage xmlns="
GetEnergyPerformanceImage_out">
  <eer>string</eer>
  <eir>string</eir>
</GetEnergyPerformanceImage>
```

Field	Description
eer	Energy Efficiency Rating (Energy consumption).
eir	Environmental Impact Rating (CO2 Emissions).

GetKeys

The GetKeys web service operation returns “id – description” pair of values.

The input XML to this operation looks like this:

```
<getKeys xmlns="GetKeys_in">
  <idre>string</idre>
  <ci>string</ci>
  <sf>boolean</sf>
  <keyTypes>
    <kt>string</kt>
    <kt>string</kt>
  </keyTypes>
  <filters>
    <sv ty="string">
      <val xmlns="SearchProperties_in">string</val>
    </sv>
    <sv ty="string">
      <val xmlns="SearchProperties_in">string</val>
    </sv>
  </filters>
</getKeys>
```

Field	Description
sf	Sale flag. Set to true to return keys only applied to properties that are for sale. (This currently only applies to the Property Type keys)
keyTypes	Array of keytypes. See Appendinx B – Key types for a list of valid key types.
filters	List of search values to filter. (Has Price and Has Photos). Refer to the Search parameters and Search Values section for more information. (This currently only applies to the Property Type keys)

Key types

The key types are specified as a two character uppercase string. As an example, getting the property types the <keyTypes /> xml would look like this:

```
<keyTypes>
  <kt>TV</kt>
```

```
</keyTypes>
```

Two examples of returned XML are shown below.

Output

The output of the operation looks like this:

```
<keys version="string" xmlns="GetKeys_out">
  <k id="int" ty="string">xml<keys><k xsi:nil="true" /><k
xsi:nil="true" /></keys><desc>string</desc><fields><fld
xsi:nil="true" /><fld xsi:nil="true" /></fields></k>
  <k id="int" ty="string">xml<keys><k xsi:nil="true" /><k
xsi:nil="true" /></keys><desc>string</desc><fields><fld
xsi:nil="true" /><fld xsi:nil="true" /></fields></k>
</keys>
```

Field	Description
id	Attribute. This is the key identifier.
ty	Attribute. Type of key.
desc	Description of key. This description is language dependent.
fld	Additional field information. See Additional fields in section about Property Details.

Example 1 – Property types

The property types in the InfoCasa system are represented in a hierarchical manner. An example of the following hierarchical representation is shown below:

- Apartments
 - Duplex
 - Studio

```
<?xml version="1.0" encoding="utf-16"?>
<keys xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="1.0" xmlns="GetKeys_out">
  <k id="1138" ty="TV">
    <keys>
      <k id="300" ty="TV">
        <keys />
        <desc>Duplex</desc>
        <fields />
      </k>
      <k id="322" ty="TV">
        <keys />
        <desc>Studio</desc>
        <fields />
      </k>
    </keys>
  </k>
```

```
</k>
</keys>
<desc>Apartments</desc>
<fields />
</k>
</keys>
```

Example 2 – Currency

The currency keys have additional information regarding the Exchange Rate in relationship to the Euro. This [additional information](#) can be seen in the example below:

```
<?xml version="1.0" encoding="utf-16"?>
<keys xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="1.0" xmlns="GetKeys_out">
  <k id="398" ty="MO">
    <keys />
    <desc>Algerian Dinar</desc>
    <fields>
      <fld tp="ER" dt="decimal">
        <val>87.4377</val>
      </fld>
    </fields>
  </k>
  <k id="396" ty="MO">
    <keys />
    <desc>British Pound</desc>
    <fields>
      <fld tp="ER" dt="decimal">
        <val>0.6843</val>
      </fld>
    </fields>
  </k>
</keys>
```

GetAreas

The GetAreas function returns areas from the database and has the following structure as input:

```
<getAreas xmlns="GetAreas_in">
  <idre>string</idre>
  <ci>string</ci>
  <filters>
    <sv ty="string">
      <val xmlns="SearchProperties_in">string</val>
    </sv>
    <sv ty="string">
      <val xmlns="SearchProperties_in">string</val>
    </sv>
  </filters>
  <sf>boolean</sf>
  <sc>string</sc>
  <pa>int</pa>
  <le>int</le>
  <sl>boolean</sl>
</getAreas>
```

Field	Description
filters	List of search values to filter. (Has Price, Has Photos and Area Group). Refer to the Search parameters and Search Values section for more information. (This currently only applies to the Property Type keys)
sf	Sale flag. Set to true to return keys only applied to properties that are for sale. (This currently only applies to the Property Type keys)
sc	Search Criteria. Letters that any part of an area starts with.
pa	Parent area.
le	Level of where to get the areas. See the Appendix C – Area levels for valid values.
sl	Sub levels flag. Set this flag to true to get sub levels. Currently gets one level down. E.g. Specify level 2 gets level 2 and 3.

Search criteria (sc)

When specifying a search criteria all other criteria are ignored. An example use of this parameter is displayed below:

Set the Search criteria (sc) value to “mar”:

It returns similar results to what is displayed below:

Banyeres de **Mariola** - Alicante
 Cuevas de San **Marcos** - Málaga
 L'Ametlla de **Mar** - Tarragona
Mar Menor - Murcia
Maracena - Granada
Marbella - Málaga
Marinaleda - Sevilla
 Roquetas de **Mar** - Almería
 San **Martín** del Tesorillo - Cádiz
 Sant **Martí** Vell - Girona
 Torre del **Mar** - Málaga
 Balcon al **Mar** - Jávea, Alicante
 Cap **Martí** - Jávea, Alicante
 El **Marquesado** - Chiclana, Cádiz
 Jardín del **Mar** - Torre Vieja, Alicante
 La **Marina** - Elche, Alicante
 Las **Marinas** - Dénia, Alicante
 Llano de las **Maravillas** - Chiclana, Cádiz
 Los **Martínez** del Puerto - Murcia, Murcia
Mar Azul - Jávea, Alicante
Mar de Cristal - La Manga, Murcia
Mar Menor Golf Resort - Mar Menor, Murcia
Marbella Alta - Marbella, Málaga
Marbella Centro - Marbella, Málaga
Marbella East - Marbella, Málaga
Marbella West - Marbella, Málaga
Maria - Vélez-Rubio, Almería
María de la Salut - Mallorca, Baleares
Marina del Este - La Herradura, Granada
Marina del Sol - Fuengirola, Málaga

As seen from the results, any part of the area that starts with the criteria selected are returned.

Output

The output xml looks like this:

```

<areas version="string" xmlns="GetAreas_out">
  <a id="int" at="string">xml
    <areas>
      <a xsi:nil="true" />
      <a .../>
    </areas>
    <ld>string</ld>
    <na>string</na>
  </a>
  <a .../>
</areas>
  
```

Field	Description
id	Attribute. This is the area identifier.
at	Attribute. Area type. This is not currently used.
ld	Long description. This contains the Town and Province of an area. Only returned when searching for areas using the Search criteria (sc) .
na	Name of area

Below is an example of the xml returned from the function:

```
<?xml version="1.0" encoding="utf-16"?>
<areas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="1.0" xmlns="GetAreas_out">
  <a id="2995" at="">
    <areas />
    <ld>Alicante</ld>
    <na>Banyeres de Mariola</na>
  </a>
  <a id="3994" at="">
    <areas />
    <ld>Málaga</ld>
    <na>Cuevas de San Marcos</na>
  </a>
  <a id="3174" at="">
    <areas />
    <ld>Tarragona</ld>
    <na>L´Ametlla de Mar</na>
  </a>
  <a id="1369" at="">
    <areas />
    <ld>Murcia</ld>
    <na>Mar Menor</na>
  </a>
  <a id="5849" at="">
    <areas />
    <ld>Granada</ld>
    <na>Maracena</na>
  </a>
</areas>
```

GetAreaGroups

The GetAreaGroups function returns area groups from the database that has been set, in the InfoCasa system, to be visible in web. It has the following structure as input:

```
<getAreaGroups xmlns="GetAreaGroups_in">
  <idre>string</idre>
  <ci>string</ci>
  <pf>boolean</pf>
  <filters>
    <sv ty="string">
      <val xmlns="SearchProperties_in">string</val>
    </sv>
    <sv ty="string">
      <val xmlns="SearchProperties_in">string</val>
    </sv>
  </filters>
</getAreaGroups>
```

Field	Description
filters	List of search values to filter. (Has Price and Has Photos). Refer to the Search parameters and Search Values section for more information.
pf	Predefined flag. Set to True to bring back groups defined by InfoCasa in which the agent has available properties. See the Appendix D – Predefined Area Groups for groups available from InfoCasa.

Output

The output xml looks like this:

```
<areaGroups version="string" xmlns="GetAreaGroups_out">
  <ag id="int">xml<na>string</na></ag>
  <ag id="int">xml<na>string</na></ag>
</areaGroups>
```

Field	Description
id	Attribute. Identifier of the Area Group.
na	Name of area group.

Below is an example of area groups returned from the operation.

```
<?xml version="1.0" encoding="utf-16"?>
<areaGroups xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0" xmlns="GetAreaGroups_out">
  <ag id="-1002">
    <na>Costa Almería</na>
  </ag>
  <ag id="-1006">
    <na>Costa Azahar</na>
  </ag>
  <ag id="-1004">
    <na>Costa Blanca</na>
  </ag>
  <ag id="-1009">
    <na>Costa Brava</na>
  </ag>
  <ag id="-1003">
    <na>Costa Cálida</na>
  </ag>
  <ag id="-1010">
    <na>Costa de la Luz</na>
  </ag>
  <ag id="-1000">
    <na>Costa del Sol</na>
  </ag>
  <ag id="-1007">
    <na>Costa Dorada</na>
  </ag>
  <ag id="-1008">
    <na>Costa Maresme</na>
  </ag>
  <ag id="-1001">
    <na>Costa Tropical</na>
  </ag>
  <ag id="-1005">
    <na>Costa Valencia</na>
  </ag>
</areaGroups>
```

GetSavedSearchInfo [Deprecated]

In the InfoCasa system, it is possible to save a predefined search which could be run simply by clicking a link on a web page for example. The GetSavedSearchInfo operation returns information about what saved searches are available to an agent (i.e. saved searches stored in the system relating to that agent in particular.)

The input to this function is as following:

```
<getSavedSearchInfo xmlns="GetSavedSearchInfo_in">
  <idre>string</idre>
  <ci>string</ci>
  <type>int</type>
</getSavedSearchInfo>
```

Field	Description
type	Type of saved searches to bring back. If this is set to 0, all saved searches are brought back.

Output

The output of the operation is defined as the following:

```
<savedSearches version="string"
xmlns="GetSavedSearchInfo_out">xml
  <searches>
    <s id="int" type="int">xml
      <desc>string</desc>
    </s>
    <s .../>
  </searches>
</savedSearches>
```

Field	Description
id	Attribute. Saved search identifier. Use this identifier to get the actual search parameters saved. This is used as the input for the GetSavedSearch operation.
type	Attribute. Type of saved search.
desc	Description of saved search. This is translatable.

GetSavedSearch [Deprecated]

This function returns saved [Search parameters](#) which can be run against the InfoCasa system by the click of a button or a link on a web page.

The input of this operation looks like this:

```
<getSavedSearch xmlns="GetSavedSearch_in">
  <idre>string</idre>
  <ci>string</ci>
  <id>int</id>
</getSavedSearch>
```

Field	Description
id	Saved search identifier. This identifier is returned from the GetSavedSearchInfo operation described above.

The output of this operation are the saved [Search parameters](#) which can then be run against the [CountProperties](#) and [SearchProperties](#) operations.

GetAgencies

This operation returns the details of InfoCasa agencies sharing properties with the licence.

The input of this operation looks like this:

```
<getAgencies xmlns="GetAgencies_in">
  <idre>string</idre>
  <ci>string</ci>
  <idreo>string</idreo>
</getAgencies>
```

Field	Description
Idreo	Source IDRE to filter by, it could be empty to get all agencies sharing with the licence, a five characters IDRE to get the details of a specific agency or a 3 characters IDRE to get the details of the agencies in a group.

Output

The output of the operation is defined as the following:

```
<agencyDetails xmlns="Agencies_out">
  <agencies>
    <agency>xml
      <idre>string</idre>
      <boss>boolean</boss>
      <name>string</name>
      <web>string</web>
      <hlnk>string</hlnk>
      <email>string</email>
      <phone>string</phone>
      <fax>string</fax>
      <area id="int">
        <name>string</name>
        <ada>string</ada>
        <country>int</country>
      </area>
      <address>string</address>
    </agency>
    <agency> ... </agency>
  </agencies>
</agencyDetails>
```

Field	Description
idre	Agency IDRE
boss	Is agency a Boss ?
name	Name of the agency
web	Web site of the agency
hlnk	Property details hiperlink
email	Agency e-mail
phone	Agency telephone
fax	Agency fax
area	Location element
area (id attribute)	Area Id.
area (name element)	Area name
area (ada element)	Area Description Ascending
area (country element)	Country Id. of the area
address	Address / specific area

WebUser [Deprecated]

The details of a web user are used as input or output of several of the WebUser operations.

The XML of a WebUser looks like:

```
<webUser xmlns="webUser">
  <idre>string</idre>
  <ci>string</ci>
  <userId>int</userId>
  <email>string</email>
  <pwd>string</pwd>
  <title>string</title>
  <name>string</name>
  <sname>string</sname>
  <homePhone>string</homePhone>
  <workPhone>string</workPhone>
  <fax>string</fax>
  <mobilePhone>string</mobilePhone>
  <na>int</na>
  <co>int</co>
  <birthday>date</birthday>
  <autoMail>boolean</autoMail>
  <source>int</source>
  <notes>string</notes>
</webUser>
```

Field	Description
userId	User Id.
title	Title
name	Name
sname	Surname
homePhone	Home telephone number
workPhone	Work telephone number
fax	Fax telephone number
mobilePhone	Mobile telephone number
na	Nationality Id
co	Country Id
birthday	Birthday Date
autoMail	Wants to receive automatic emails.
source	Advert Source Id.
notes	Notes about the user.

RegisterWebUser [Deprecated]

The RegisterWebUser operation is called to register a new WebUser.

The input and output of this operation is of type WebUser described in the previous section.

The new user is created with status PENDING_ACTIVATION and an activation e-mail is sent to the e-mail address provided.

Output

The output xml would look like this:

```
<RegisterWebUserResponse xmlns="urn:schemas-infocasa-com:client-
service:2005-11">
  <webUser xmlns="webUser">
    <idre>string</idre>
    <ci>string</ci>
    <userId>int</userId>
    <email>string</email>
    <pwd>string</pwd>
    <title>string</title>
    <name>string</name>
    <sname>string</sname>
    <homePhone>string</homePhone>
    <workPhone>string</workPhone>
    <fax>string</fax>
    <mobilePhone>string</mobilePhone>
    <na>int</na>
    <co>int</co>
    <birthday>date</birthday>
    <autoMail>boolean</autoMail>
    <source>int</source>
    <notes>string</notes>
  </webUser>
</RegisterWebUserResponse>
```

Here is an example of a webuser returned from the operation:

```
<RegisterWebUserResponse xmlns="urn:schemas-infocasa-com:client-service:2005-11">
  <webUser xmlns="webUser">
    <idre>JOSEC</idre>
    <ci>en</ci>
    <userId>55</userId>
    <email>peter.piper@gmail.es</email>
    <pwd>mypassword</pwd>
    <title>Mr</title>
    <name>Peter</name>
    <sname>Pipper</sname>
    <homePhone>952666666</homePhone>
    <workPhone>666666666</workPhone>
    <fax>952666666</fax>
    <mobilePhone>666666666</mobilePhone>
    <na>500</na>
    <co>533</co>
    <birthday>1975-03-14T00:00:00</birthday>
    <autoMail>true</autoMail>
    <source>1138</source>
    <notes>bla bla bla</notes>
```

```
</webUser>
</RegisterWebUserResponse>
```

ActivateWebUser [Deprecated]

Once the web user has been registered, he needs to be activated before he can log in; this is done via a confirmation email that is automatically sent to the email address the user registered with.

The operation that activates a user so that he can log in to his account is the ActivateWebUser operation.

The operation input would be:

```
<activateWebUser xmlns="ActivateWebUser_in">
  <idre>string</idre>
  <ci>string</ci>
  <userId>int</userId>
  <activationCode>string</activationCode>
</activateWebUser>
```

Field	Description
userId	User Id (returned by RegisterWebUser operation)
activationCode	Activation code the user has received

Output

The output of the operation is:

```
<activateWebUser xmlns="ActivateWebUser_out">
  <resultCode>int</resultCode>
  <resultMsg>string</resultMsg>
</activateWebUser>
```

Field	Description
resultCode	Result code. 1 if successful
resultMsg	Descriptive text for the result code.

LoginWebUser [Deprecated]

This operation will authenticate the login details of a user. The input parameters are defined as follows:

```
<logInWebUser xmlns="LogInWebUser_in">
  <idre>string</idre>
  <ci>string</ci>
  <email>string</email>
  <pwd>string</pwd>
</logInWebUser>
```

Field	Description
email	User email address
pwd	User password

Output

The output of the operation is of type webUser.

Here is an example of a webUser returned after the operation:

```
<webUser xmlns="webUser">
  <idre />
  <ci>en</ci>
  <userId>56</userId>
  <email>peter.pipper@gmail.com</email>
  <title>Sr</title>
  <name>Peter</name>
  <sname>Pipper</sname>
  <homePhone>555555555</homePhone>
  <workPhone>555555555</workPhone>
  <fax>555555555</fax>
  <mobilePhone>555555555</mobilePhone>
  <na>0</na>
  <co>0</co>
  <birthday>2009-03-02T00:00:00</birthday>
  <autoMail>false</autoMail>
  <source>0</source>
  <notes />
</webUser>
```

ModifyWebUser [Deprecated]

Operation that modifies the details of a web user. This operation receives a [WebUser](#) as input.

Output

The output would be:

```
<modifyWebUser xmlns="ModifyWebUser_out">
  <resultCode>int</resultCode>
```

```
<resultMsg>string</resultMsg>
</modifyWebUser>
```

Field	Description
resultCode	Result code of the operation. 1 if successful or an error code if not.
resultMsg	Descriptive text for the result of the operation.

ModifyWebUserPwd [Deprecated]

Operation that modifies the password of a web user. This operation receives a modifyWebUserPwd as input.

```
<modifyWebUserPwd xmlns="ModifyWebUserPwd_in">
  <idre>string</idre>
  <ci>string</ci>
  <userId>int</userId>
  <email>string</email>
  <pwd>string</pwd>
  <newPwd>string</newPwd>
</modifyWebUserPwd>
```

Field	Description
userId	User id.
email	The user email address.
pwd	Current user password.
newPwd	New password for the user.

Output

The output would be of type modifyWebUser, explained as the [output](#) of the ModifyWebUser operation in the previous section.

RememberWebUserPwd [Deprecated]

Operation used to remind the password to a user. If the operation is successful an email is sent to the email address the user registered with, containing a newly generated password.

```
<rememberWebUserPwd xmlns="RememberWebUserPwd_in">
```

```

    <idre>string</idre>
    <ci>string</ci>
    <email>string</email>
  </rememberWebUserPwd>

```

Field	Description
email	The email address of the user.

Output

The output would be of type ModifyWebUser, explained as the [output](#) of the ModifyWebUser operation in the previous section.

Example output:

```

<modifyWebUser xmlns="ModifyWebUser_out">
  <resultCode>1</resultCode>
  <resultMsg>Successful</resultMsg>
</modifyWebUser>

```

AddPropertyEnquiry

In order to use this operation, agency must be using InfoCasa web based application. If you are still using the old desktop application, please contact support to migrate to the web based one.

This operation will accomplish two tasks:

1. Create a contact in InfoCasa, assigned to the specified user and source. If there were a contact with the email provided, no contact is created.
2. Create a task of type "Lead" (including a reminder by email), associated with the contact created or found in previous point. Any properties sent will be associated with the task. If the contact already exists, the task will be assigned to the sales agent of the existing contact instead of the "assignedTo" parameter received.

The contact is created with these details:

Name

Surname

Email addresses

Telephone numbers

Source

Assigned to

Preferred language

Register by: All contacts created through the web service appear created by a system user called "Website integration".

The task is created with these details:

Created by: "Website integration" system user.

Assigned to

Title

Type: Lead

Contact

Property

Content: Text content of the request, including text and additional fields provided.

The input parameters are defined as follows:

```
<addPropertyEnquiry xmlns="addPropertyEnquiry_in">
  <idre>string</idre>
  <ci>string</ci>
  <ids>
    <id>int</id>
    <id>int</id>
  </ids>
  <text>string</text>
  <assignedTo>int</assignedTo>
  <name>string</name>
  <surname>string</surname>
  <emails>
    <email ty="string" />
    <email ty="string" />
  </emails>
  <telephones>
    <telephone ty="string" />
    <telephone ty="string" />
  </telephones>
  <source>int</source>
  <preferredLanguage>string</preferredLanguage>
  <addFields>string</addFields>
</addPropertyEnquiry>
```

Field	Description
-------	-------------

idre	See Common xml elements
ci	See Common xml elements
ids	Array of property identifiers. Optional. If any id is sent, the task created will be associated with those properties.
text	Text / notes field filled by the end user in the web form. Max. length is 300 characters.
assignedTo	User id the contact and task will be assigned to. Ask InfoCasa for the id corresponding to the user you would like to use.
name	End user's name filled in the web form.
surname	End user's surname filled in the web form.
emails	List of end user's email addresses with their corresponding type. At least one email must be specified. Valid types are: work, home, other.
telephones	List of end user's telephone numbers with their corresponding type. Optional. Valid types are: work, home, mobile, fax, other.
source	Id of the source which will be applied to the contact created. Optional. Ask InfoCasa for the id corresponding to the source you would like to use.
preferredLanguage	Preferred language to apply to the created contact, Optional. if not specified, same value than "ci" will be assumed.
addFields	Optional. If you collect any other information from the end user in your web form, you can specify them in this field. The content of this field will be saved in the content of the task created. Max. length is 300 characters.

Output

The output will only contain a Boolean field indicating if the operation could be executed successfully.

Example output:

```
<addPropertyEnquiry xmlns="AddPropertyEnquiry_out">  
  <resultCode>boolean</resultCode>  
</addPropertyEnquiry>
```

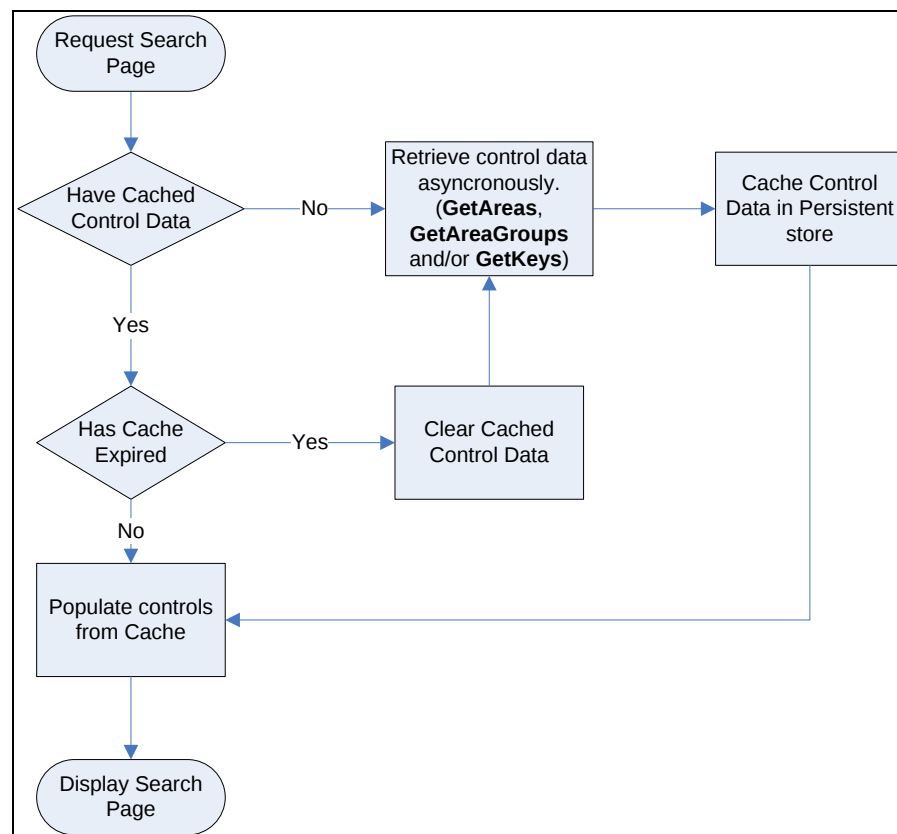
Example work flows

The following section explains best practice work flows for calling the web services from a typical web site.

Note: In the diagrams below the text outlined in **bold** indicates calls to the Client Web Service.

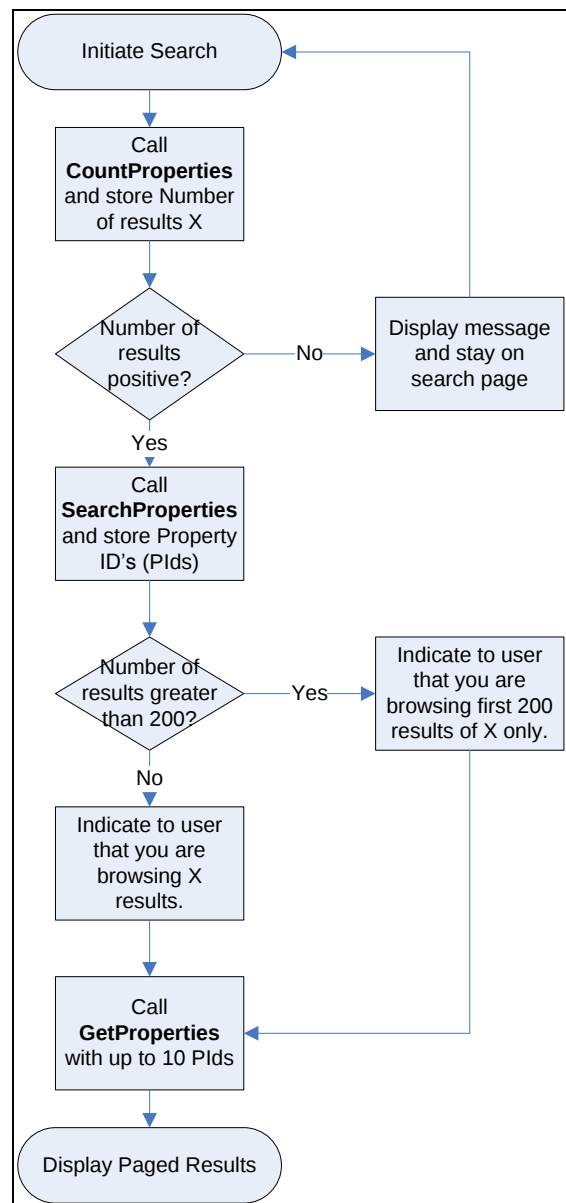
Building a search page

The first thing you would want when designing a search engine for your clients to be able to find properties through the InfoCasa system would be a Search Page. The **GetAreas**, **GetAreaGroups** and **GetKeys** controls provide access to data that you might want to display on a search page. Below is a flow diagram of the recommended way of loading the data for a search page including calls to the Web Services.



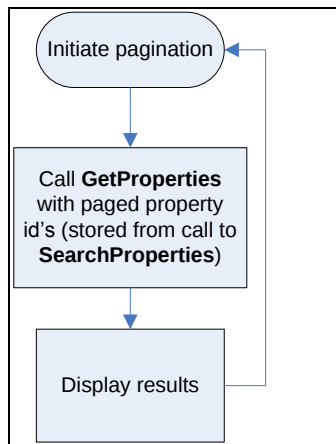
Initialising a search

Once the user has selected desired search criteria and hit the Search button, the following flow diagram describes the best practice for handling the search initialisation:



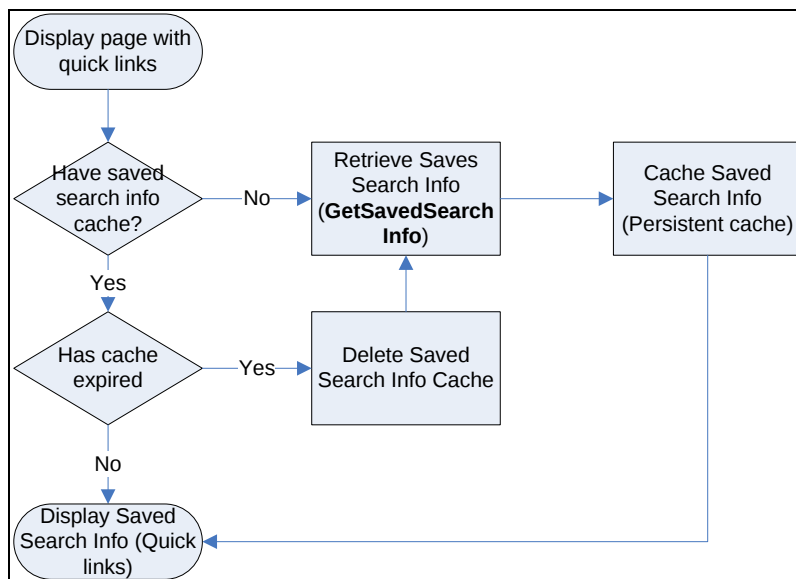
Pagination

Once a search has been initiated and the first page displayed, the following workflow is recommended to page through properties:

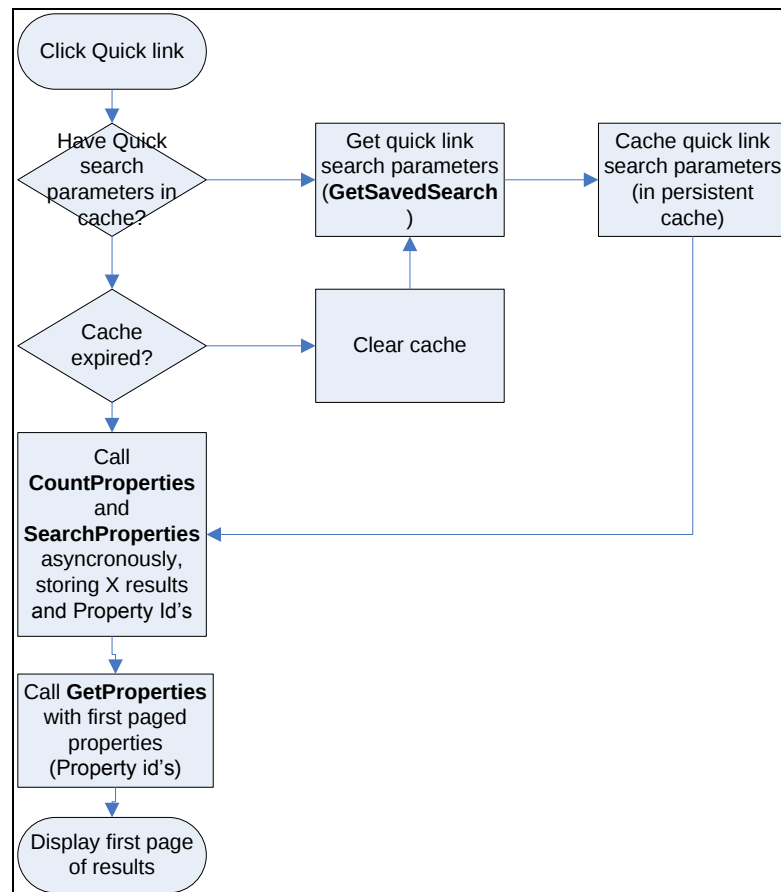


Quick links

If you want to add direct links to predefined searches, the InfoCasa system provides you with SavedSearches. To display these saved searches on a page, follow the following flow diagram below:



Once these quick links have been displayed, the following work flow is recommended to initialise the actual saved search.



Property details

Once details of a property has been displayed in a list from either a Quick link, or a property search the **GetPropertyDetails** operation can be called to display full details of a property from the InfoCasa system.

Appendix A – Country identifiers

Spain = 3,
South Africa = 1115,
Germany = 1116,
Ireland = 1142,
Gibraltar = 1245,
USA = 1370,
Croatia = 1614,
Argentina = 2361,
Portugal = 2395,
France = 2719,
Thailand = 3041,
Turkey = 3102,
Morocco = 3146,
Czech Republic = 3147,
Hungary = 3148,
United Kingdom = 3300,
Cyprus = 3668,
Letonia = 4172,
Estonia = 4236,
Malta = 4237,
Brasil = 4238,
Austria = 4239,
Greece = 4240,
Italy = 4262,
Indonesia = 4265,
Andorra = 4415,
United Arab Emirates = 4432,
Northern Cyprus = 4507,
Romania = 4650,
Bulgaria = 4651,
Dominican Republic = 4907,
Panama = 4912,
India = 5739,
Bahamas = 5770,
Venezuela = 5796,
Cape Verde = 5797

Note: The GetAreas and GetAreaGroups expect a filter value of type Country (C). It can however be set to 1 to ignore the country.

Appendix B – Key types

Country = CO,
Features = FE,
Furnished = FU,
Garden = JA,
Kitchen = KI,
Currency = MO,
Orientation = OR,
Parking = PA,
Pool = PO,
Location = TU,
Property Type = TV,
Views = VI

Appendix C – Area levels

Country = 1,
Province = 2,
Town = 3

Appendix D – Predefined Area Groups

The following area groups are predefined by InfoCasa:

- Costa Almería
- Costa Azahar
- Costa Blanca
- Costa Brava
- Costa Cálida
- Costa de la Luz
- Costa del Sol
- Costa Dorada
- Costa Maresme
- Costa Tropical
- Costa Valencia

Appendix E – Currency codes

Euro = 394,

Pesetas = 395,
British Pound = 396,
US Dollar = 397,
Algerian Dinar = 398,
Cuban Peso = 399,
Italian Lira = 400,
Japanese Yen = 401,
German Mark = 402,
Danish Krone = 403,
French Franc = 404,
Indian Rupee = 405,
Portuguese Escudo = 406,
Russian Rouble = 407,
Dutch Guilder = 408,
Swedish Krona = 481,
Norwegian Kroner = 482,
South African Rand = 1054,
Iceland Krona = 1067,
Swiss Franc = 1076,
Cyprus Pound = 1103,
Maltese Lira = 1104,
Moroccan Dirham = 1144,
United Arab Emir. Dirham = 1175,
Czech Koruna = 1181,
Belgian Franc = 1183,
Irish Punt = 1184,
Argentine Peso = 1224

Appendix F – Rental Price Frequency values

Daily = 1453,
Weekly = 1454,
2 Weeks = 1455,
Monthly = 1456,
2 Months = 1457,
Quarterly = 1458,
4 Months = 1459,
5 Months = 1460,
6 Months = 1461,
Yearly = 1462

Appendix G – Property status values

Live - Valuation = 1,
Live - Available = 2,
Live - Offer Made = 3,
Live - Reserved = 4,
Live - Empty = 9,
Live - Rented = 13,
Live - Sold = 14,
Live - Subject To Contract = 15,

Archived - Temporary = 5,
Archived - Sold = 6,
Archived - Withdrawn = 7,
Archived - Expired = 8,
Archived - Empty = 10,
Archived - Rented = 16,
Archived - Rejected = 17,
For Authorization = 11,
Rejected = 12

Appendix H – Getting properties images

Images of properties can be easily obtained through the following URL:

```
https://cdn.infocasa.com/property-image/{width}_{property_image}
```

Parameter	Description
width	Desired image width. Following values are available at the moment: • small (150 px) • medium (300 px) • large (640 px) • xlarge (1024 px) • xxlarge (1300 px)
property_image	Image name. This name is returned in the attachments section of GetProperties and GetPropertyDetails operations.