Xpressdocs Listing Feed Integration Interface

May 2021

Xpressdocs Holdings, Inc. 1301 NE Loop 820, Fort Worth, TX 76131, USA
+1 817.321.7904 | https://www.xpressdocs.com/xd

Table of Contents

1 Introduction

This document describes the Xpressdocs Listing Feed Integration Interface. A customer can implement this interface as an HTTP(S) API to allow Xpressdocs to retrieve current listing data into the Xpressdocs platform. Xpressdocs will make this data available to the end users of the customer. The data may be utilized throughout the Xpressdocs platform for a variety of products and services, including: Automated Property Marketing Program, Xpress Connection Program, Email Marketing, My Listings, Xpress Mobile and other services.

2 Security

This API relies on HTTP(s) pull requests for retrieval of data feed. The customers HTTP(s) end points can be secured using HTTP Basic/Digest authentication schemes, we also support OAUTH2 authentication.

3 Listing Feed Integration

3.1 Overview

Xpressdocs expects the API to provide listing data in JSON format specified in the subsequent sections of this document. This data would include all pertinent property data, dates such as listing, openhouse and sold date, agent data and any image URLs associated with the listing. Xpressdocs will retrieve data at set intervals throughout the day, these intervals could be as short as every 5 minutes.

3.1.1 Associated Photos

For best quality, associated photo URLs should reference the image directly and not an image manipulation utility or other image proxy. If a proxy is required, the quality should be verified early in the integration process; high-resolution images are preferred.

When Xpressdocs imports associated photos, they are imported in the same order provided in the data feed.

4 Data Feed Data Fields

This section describes the structure of the JSON data feed in detail.

4.1 API parameters

Xpressdocs requires that the API support delta pulls. The API client can specify the offset and limit of listings data in the response.
The following parameters should be supported by the API URL.

Parameter	Data Type	Required	Description	Notes
fromDate	ISODate format timestamp	Yes	Used to limit listings returned to only those last modified after the fromDate	
toDate	ISODate format timestamp	No	Used to limit listings to only those modified before the toDate	This field won't normally be used unless preloading old data.
limit	number	Yes	Used for paginating results.	Xpressdocs would generally limit this to 100
offset	number	Yes	Used for paginating results.	We will cycle through this until there are no results.
listingId	String	No	Customers unique id reference for listing. When this filter is used, we expect only one listing to be returned if it exists.	

4.2 Data Feed Structure

The customers API's response should be a JSON object with a field name called "listings" at the top level. Its value will be an array of listings objects.

Example:

```
{
   "listings": [
      {}
   ]
}
```

4.3 Data Feed Schemas

Each listing will be its own object. The object will include the fields described below.

```
{
   "location": {
   // Object described under Location schema data fields section
   },
   "agent": {
   // Object described under Agent schema data fields section
   },
   "openHouse": {
   // Object described under Open House details schema data fields section
   },
   "pictures": {
   // Object described under Pictures Schema data fields section
   },
   "buyerSeller": {
   // object is described under Buyer/Seller Schema data fields section
   },
   // Additional properties at the top level of the object
   // Described under the Listings details fields section
}
```

All the fields to be included, whether they are required, the Xpressdocs programs/features for which they are frequently used, and the default template field associated with each field are outlined below.

4.3.1 Location Schema Data Fields

The following data elements should be included in the Location Object.

```
"location": {
      "streetAddress": "",
      "city": "",
      "state": "",
      "zip": "",
      "unitNumber": "",
      "lat": "",
    "long": ""
  }
```

Field	Data Type	Description	Required	Applicable Program(s)	Template Field
streetAddress	String	Property Location	Yes	All	street address
unitNumber	String	Property Unit Number	No	All	unit or suite
city	String	Property City	Yes	All	city
state	String	Property State	Yes	All	state
zip	String	Property Postal Code	Yes	All	zip or postal code
lat	String	Property Latitude	No	Email Marketing	N/A
long	String	Property Longitude	No	Email Marketing	N/A

4.3.2 Agent Schema Data Fields

The following fields should be included in the Agent object.

```
"agent": {
    "firstName": "",
    "lastName": "",
    "userId": "",
    "emailAddress": ""
  }
```

Field	Data Type	Description	Required	Applicable Program(s)
firstName	String	First Name of Associated Employee/Agent	Yes	All
lastName	String	Last Name of Associated Employee/Agent	Yes	All
userID	String	Unique Identifier of Associated Employee/Agent	Yes	All

emailAddress	String	Email Address of Associated Employee/Agent	Yes	All

4.3.3 Open House Details Schema Data Fields

The Open House field will be an array of objects. The following data fields should be included in each Open House Object.

```
"openHouse": [

    {
      "date": "2020-02-08",
      "startTime": "10:00:00-0600",
      "endTime": "17:00:00-0600"
    }
  ],
```

Field	Data Type	Description	Required	Applicable Program(s)
date	String (YYYY-MM-DD)	Date of the open house	No	APM
startTime	String (HH:MM:SS ZZ)	Start time of the open house with time offset	No	APM
endTime	String (HH:MM:SS ZZ)	End time of the open house with time offset	No	APM

4.3.4 Pictures Schema Data Fields

The Pictures field will be an array of objects. The following data fields should be included in each Pictures Object.

```
"pictures": [
  {
    "pictureUrl": "",
    "thumbUrl": ""
  }
],
```

Field	Data Type	Description	Required	Applicable Program(s)
pictureUrl	String	Associated Photos URL	No	APM, Email Marketing
thumbUrl	String	Associated Photos thumbnail URL	No	APM, Email Marketing

4.3.5 Buyer/Seller Schema Data Fields

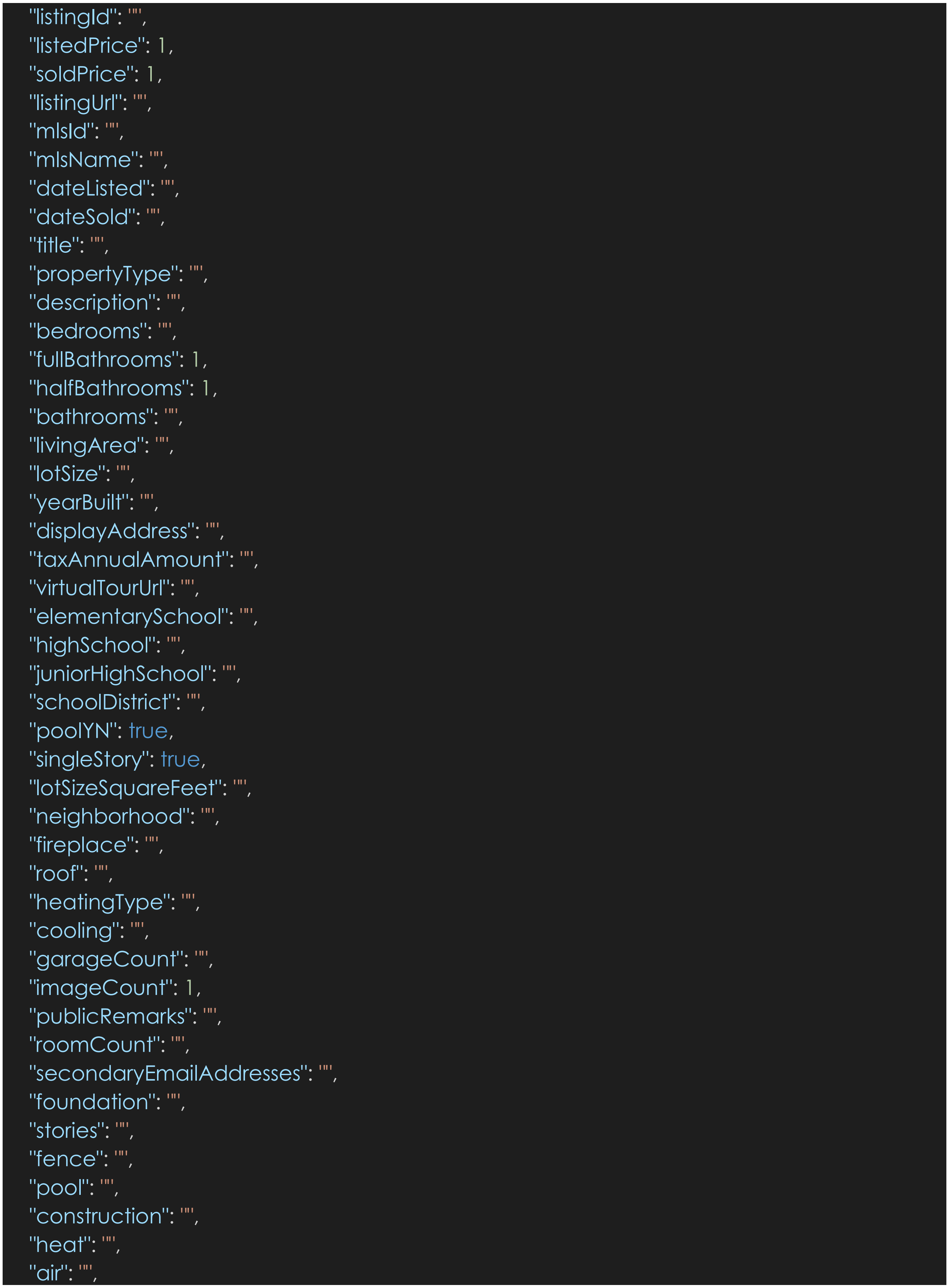
The BuyerSeller field will be an array of four objects. The first object will contain the primary buyers first name and the primary buyers' last name. The second object will include the primary sales associate first name and the primary sales associate last name. The third object will contain the secondary buyers first name and the secondary buyers' last name. The fourth object will include the secondary sales associate first name and the secondary sales associate last name.

```
"buyerSeller": [
      {
         "primaryBuyerFirstName": "",
         "primaryBuyerLastName": ""
      },
      {
         "primarySalesAssociateFirstName": "",
         "primarySalesAssociateLastName": ""
      },
      {
         "secondaryBuyerFirstName": "",
         "secondaryBuyerLastName": ""
      },
      {
         "secondarySalesAssociateFirstName": "",
         "secondarySalesAssociateLastName": ""
      }
  ]
```

Field	Data Type	Description	Required	Applicable Program(s)
primaryBuyerFirstName	String	First name of primary buyer	No	XC
primaryBuyerLastName	String	Last name of primary buyer	No	XC
primarySalesAssociateFirstName	String	First name of primary sales associate	No	XC
primarySalesAssociateLastName	String	Last name of primary sales associate	No	XC
secondaryBuyerFirstName	String	Last name of primary buyer	No	XC
secondaryBuyerLastName	String	Last name of primary buyer	No	XC
secondarySalesAssociateFirstName	String	First name of secondary sales associate	No	XC
secondarySalesAssociateLastName	String	Last name of secondary sales associate	No	XC

4.3.6 Listing Details Fields

The following fields should be placed at the top level inside the listing object, alongside the fields that were previously discussed in 4.2.1 – 4.2.



```
"listingId": "",
"listedPrice": 1,
"soldPrice": 1,
"listingUrl": "",
"mlsId": "",
"mlsName": "",
"dateListed": "",
"dateSold": "",
"title": "",
"propertyType": "",
"description": "",
"bedrooms": "",
"fullBathrooms": 1,
"halfBathrooms": 1,
"bathrooms": "",
"livingArea": "",
"lotSize": "",
"yearBuilt": "",
"displayAddress": "",
"taxAnnualAmount": "",
"virtualTourUrl": "",
"elementarySchool": "",
"highSchool": "",
"juniorHighSchool": "",
"schoolDistrict": "",
"poolYN": true,
"singleStory": true,
"lotSizeSquareFeet": "",
"neighborhood": "",
"fireplace": "",
"roof": "",
"heatingType": "",
"cooling": "",
"garageCount": "",
"imageCount": 1,
"publicRemarks": "",
"roomCount": "",
"secondaryEmailAddresses": "",
"foundation": "",
"stories": "",
"fence": "",
"pool": "",
"construction": "",
"heat": "",
"air": "",
  "block": "",
  "levels": ""
```

Field	Data Type	Description	Required	Applicable Program(s)	Template Field
status	String	Property Status (pending, closed, available, sold)	Yes	All	N/A
listingId	String	Unique ID for the listing	No		
listedPrice	String	Sale Price	No	APM, Email Marketing	Price
soldPrice	String	Sold Price	No	APM, Email Marketing	Sold Price
listingUrl	String	Listing URL	No	All	Listing URL
mlsId	String	Property Listing Unique Identifier	Yes	All	mls
mlsName	String	Name of Listing Service	Yes	All	N/A
dateListed	String (ISO DateTime Format)	Property Listing Date	Yes	APM, Email Marketing	Date Listed
dateSold	String (ISO DateTime Format)	Property Sold Date	No (required for some programs or if DateListed is not provided)	All	Date Sold
title	String	Title of Marketing	No	APM, Email Marketing	Headline or Title
propertyType	String	Type of Property (Single Family, Residential, Condo, Lots & Land)	No	APM, Email Marketing	Property Type

description	Text	Description of Property	No	APM, Email Marketing	Description
bedrooms	String	Number of Bedrooms	No	APM, Email Marketing	Bedrooms
fullBathrooms	String	Number of Full Bathrooms	No	APM, Email Marketing	N/A
halfBathrooms	String	Number of Half Bathrooms	No	APM, Email Marketing	N/A
bathrooms	String	Number of Bathrooms	No	APM, Email Marketing	Bathrooms
livingArea	String	Number of Living Areas or Dimensions of Living Area	No	APM, Email Marketing	Living area or Square

Field	Data Type	Description	Required	Applicable Program(s)	Template Field
lotSize	String	Property Lot Size	No	APM, Email Marketing	Lot Size
yearBuilt	String	Year Property was Built	No	APM, Email Marketing	Year built or year
DisplayAddress	String	True/False indication of whether address should be displayed	No	Email Marketing	N/A
taxAnnualAmount	String	Amount of estimated taxes	No	APM	Taxes
virtualTourUrl	String	Virtual Tour URL	No	APM, Email Marketing	Virtual Tour URL
elementarySchool	String	Name of elementary school	No	APM	Elementary
highSchool	String	Name of high school	No	APM	High school
juniorHighSchool	String	Name of junior high school	No	APM	Middle
schoolDistrict	String	Name of school district	No	APM	Schools
poolYN	Boolean	Pool description	No	APM	Pool
singleStory	Boolean	Single story	No	APM	N/A
lotSizeSquareFeet	String	Square feet of lot size	No	APM	N/A
neighborhood	String	Name of subdivision or neighborhood	No	APM	Subdivision

Field	Data Type	Description	Required	Applicable Program(s)	Template Field
fireplace	String	Fireplace Description	No	Email Marketing	Fireplace
roof	String	Type of roof	No	APM	Roof
heatType	String	Type of heating system	No	APM	Heat
cooling	String	Description of cooling system	No	APM	N/A
garage	String	Dimensions of Garage	No	APM	Garage
imageCount	Integer	Count of images	No	APM	N/A
publicRemarks	String	Public remarks	No	APM	N/A
roomCount	String	Number of rooms	No	APM	N/A
secondaryEmailAddress	String	Secondary Email Address	No	APM	**N/A**
foundation	String	Type of foundation	No	APM	Foundation
stories	String	Number of stories	No	APM	Stories
fence	String	Type of fence	No	APM	Fence
pool	String	Pool description	No	APM	Pool
construction	String	Type of home construction	No	APM	Construction
heat	String	Type of heating system	No	APM	Heat

5 Deployment

5.1 Environments

Xpressdocs has the following:

Name	Description
Stage	Staging environment for QA testing
Production	Live production environment

Xpressdocs will provide one non-production environment for testing purposes (Stage). A client's production environment will point to Xpressdocs production environment by default.

5.2 Tasks

The following tasks are required to complete a data feed integration:

1. Provide an API endpoint where Xpressdocs can fetch listing data in the above format.
2. Xpressdocs will validate the data feed response contains all required elements and is well-formed JSON.
3. Xpressdocs will notify client that the data feed import can begin.