

Matrix RETS Introduction

Sample

Downloading updates from the Matrix RETS server.

Some basics:

RETS is the Real Estate Transaction Standard. The definitive web site is <http://rets.org/>

We use RETS with Matrix to provide a data querying facility to our Matrix customers and their authorized third-party partners. RETS data transfers are “data pull”, initiated and controlled by the customer or their authorized partners.

Our system is a RETS 1.72 compliant system. The RETS 1.72 specification document may be located at http://www.reso.org/assets/RETS/Specifications/rets_1_7_2.pdf

Here is the example we generally give clients to try so that they can get a feel for how our RETS server works. This example uses internet explorer (IE) as a basic RETS client. **RETS is not a website but** is based on the same technology which is why we can use IE as a basic client. There are other RETS clients available in the market. (Check out <http://www.retsconnector.com/> for CoreLogic’s RETS application.)

Exploring Matrix RETS:

- Open up a brand new IE window.
- Go to <https://matrixrets.realcomponline.com/rets/login.ashx>
- IE will prompt you for the username and password. Enter them as provided by Realcomp, and don’t forget that they are case sensitive.
- IE should present an .xml formatted response.
- Change the address in the address bar to <https://matrixrets.realcomponline.com/rets/login.ashx> (note - this link is NOT clickable from this document because you must be in the same browser session that you logged on with above). This page is a holdover from the RETS 1.5 specification. On this page we have added a few example links to get some of the metadata that describes what is available in our server.

The System link

(<https://matrixrets.realcomponline.com/rets/GetMetaData.ashx?TYPE=METADATA-SYSTEM&ID=0&Format=COMPACT>) simply gives the system description.

Resources are Property, Agent, Office and OpenHouse.

Classes are the subdivisions of Resources. For the Property Resource, RESI and COMM are two Classes.

Finally, Tables gives the field makeup of each Class:

Matrix RETS Introduction

Sample

```

<?xml version="1.0"?>
- <RETS ReplyText="Operation Success." ReplyCode="0">
  - <METADATA>
    + <METADATA-TABLE Date="Tue, 23 Jul 2013 16:46:16 GMT" Version="1.00.00382" Class="RESI" Resource="Property">
    + <METADATA-TABLE Date="Mon, 22 Jul 2013 18:19:03 GMT" Version="1.00.00307" Class="LAND" Resource="Property">
    + <METADATA-TABLE Date="Tue, 23 Jul 2013 16:46:29 GMT" Version="1.00.00363" Class="INCOME" Resource="Property">
    + <METADATA-TABLE Date="Thu, 27 Jun 2013 17:19:38 GMT" Version="1.00.00040" Class="Agent" Resource="Agent">
    - <METADATA-TABLE Date="Mon, 22 Jul 2013 23:20:16 GMT" Version="1.00.00038" Class="Office" Resource="Office">
      - <Field>
        <SystemName>HeadOffice_MUI</SystemName>
        <StandardName/>
        <LongName>Head Office MUI</LongName>
        <DBName>R37</DBName>
        <ShortName>Head Office MUI</ShortName>
        <MaximumLength>19</MaximumLength>
        <DataType>Long</DataType>
        <Precision/>
        <Searchable>1</Searchable>
        <Interpretation>Number</Interpretation>
        <Alignment>Right</Alignment>
        <UseSeparator>0</UseSeparator>
        <EditMaskID/>
        <LookupName/>
        <MaxSelect/>
        <Units/>
        <Index>1</Index>
        <Minimum>0</Minimum>
        <Maximum>9223372036854775807</Maximum>
        <Default/>
        <Required>0</Required>
        <SearchHelpID/>
        <Unique>0</Unique>
      </Field>

```

The last link on the Action page is an example search that gets all of the residential listings with 9 or more beds. Note that this query uses STANDARD names, as indicated by &StandardNames=1 in

[https://matrixrets.realcomponline.com/rets/search.ashx?CLASS=Residential-Property&searchtype=Property&querytype=DMQL2&Query=\(BEDS=9%2B\)&StandardNames=1&Format=COMPACT-DECODED](https://matrixrets.realcomponline.com/rets/search.ashx?CLASS=Residential-Property&searchtype=Property&querytype=DMQL2&Query=(BEDS=9%2B)&StandardNames=1&Format=COMPACT-DECODED)

**** Depending on the specific system configuration, the hardcoded URL on the bottom of the action.ashx may or may not function as intended.**

For all your test queries, use StandardNames=0 and use the SystemName fields. This example searches for 9+ bedrooms. Note the Class name must also be "Standard"



The screenshot shows a browser address bar with the following URL: `http://matrixrets.recolorado.com/rets/search.ashx?CLASS=Resi&searchtype=Property&querytype=DMQL2&Query=(BedsTotal=9%2B)&StandardNames=0&`. Below the URL, an XML snippet is shown with red lines connecting the URL parameters to the corresponding XML elements: `<RETS ReplyText="Operation Success." ReplyCode="0">`, `- <METADATA-TABLE Date="Tue, 23 Jul 2013 16:46:16 GMT" Version="1.00.00382" Class="RESI" Resource="Property">`, and `<COLUMNS> SystemName StandardName LongName DBName ShortName MaximumLength DataType Precision`.

Matrix RETS Introduction

Sample

```
<RETS ReplyText="Operation Success." ReplyCode="0">
  <DELIMITER value="09"/>
  <COLUMNS> BathsFull BathsHalf BathsThreeQuarter BedsTotal City Conditional
    LotSizeDimensions MatrixModifiedDT MiddleOrJuniorSchool MLS MLSNumber
    PublicRemarks SellingAgentMLSID SqftTotal StateOrProvince Status Street
  <DATA> 9 Denver Adams Westminster 043371 2007-08-13T00:00:00.000 1875
    Residential BANK OWNED TRI-PLEX HUGE LOT LOTS OF SQUARE FOOTAGE
  </DATA>
  <DATA> 9 Denver Denver Park Hill East 594885 2007-10-01T00:00:00.000 975
    IS A CHURCH. NEIGHBORHOOD OF $1,000,000 + HOMES.POTENTIAL CONV
    $61.00 PEF SQUARE FOOT! 100757 16109 CO Sold East 17Th 5152 Avenue
  <DATA> 9 Boulder Boulder Flatirons Boulder 15915N 2007-10-24T00:00:00.000
    80302 Residential RARE BOULDER CREEK LOCATION 75FT OF CREEKFRONT
```

To get all listings you could use a query like (status=|A) which would give you all active listings. Status is a Lookup type field; it is defined that way in our metadata. If the 'A' does not work, you may need to check the metadata to find out what the correct value for active is. These values may differ from Vendor to Vendor. All the allowable choices are in the metadata.

Replicating data from the RETS server

The following is the method we generally use to download listings from a server.

1. Download the entire set of listings.
2. Retrieve the largest MatrixModifiedDT from the listings in step 1.
3. Take that MatrixModifiedDT and roll it back a few minutes or seconds.
4. Search for all listings with a MatrixModifiedDT greater than the one in step 2.
5. Update the listings that you receive
6. From the listings you received find the largest MatrixModifiedDT
7. Take that MatrixModifiedDT and go to step 3.

This example is very generalized, and hopefully the following explanations will help to clear up a few issues.

Matrix RETS Introduction

Sample

Downloading the entire set of listings

Some accounts do not allow a limit great enough to download the entire listing set in a single pull. The following is the method we recommend to pull all of the listings for on any account.

1. Take note of the time that you started this process - it will be used later.
2. Use a query of (matrix_unique_id = 0+)&Limit=1000
3. Retrieve all those listings
4. Find the highest matrix_unique_id from those listings
5. Create a new query with (matrix_uique_id=<HighestID from step 4>+)&Limit=1000
6. Repeat from step 3 until you receive all listings.
7. Start the general download of listings using the dt_mod equal to the date you got in step 1. This makes sure to get any changes you may have missed while getting the entire set of listings.

This process works because all of our output is ordered by the primary key, the Matrix_unique_Id field. The RETS spec does not allow a client to specify an order.

Keeping up to date with removed/non-visible listings.

Listings can and do disappear from an account's visibility. Deleted listings, zapped listings, listings in a status that an account is not allowed to see, restricted listings. For example, for an account that is only able to see active listings, a listing that goes pending or sold will seemingly vanish. The reverse is also true - if a listing goes from pending back to active, it will reappear, although it would have a modification timestamp change, and should be picked up by any ordinary update process.

The solution to this problem at the moment is to refresh the listings on a regular interval. We suggest that a process like this be run nightly.

1. Perform a RETS search like this (matrix_unique_ID=0+)&Select=matrix_unique_ID
2. This should return a list of all of the matrix_unique_id that you are allowed to see.
3. Compare this list to your current database. If you have a listing that is not in the list that you have downloaded, then you need to remove it. If there is a matrix_unique_id in the list that you do not have you need to download it. Listings can appear in your output without a modification timestamp change if there are changes to the security levels, or if there are timing restrictions like not being able to see listings until after they have been in the system for 3 days.

This method will be much faster and more bandwidth friendly than simply refreshing the entire listing set every night.

Matrix RETS Introduction

Sample

The COUNT = 1 Parameter

Using Count =1 puts a count of the listings that are going to be returned at the top of the output file. Using this parameter causes our server to have to count its listings before it starts sending them. Doing this count causes extra load on the system when generally the count variable at the top of the file is not used for anything. *If the count at the top of the file is not absolutely needed, removing the count parameter or setting it to 0 (which means no count) greatly decreases the load on the server.*

Downloading Photos

Downloading photos in RETS is done through the GetObject request. Here is an example of the GetObject request to get an image for a property. It is the code to retrieve an actual photo for MLS® ID 1177249B in the SWFL database:

<http://matrixrets.recolorado.com/rets/Action=/GetObject.ashx?Type=Photo&Resource=Property&ID=11738756:0>

Note - ID is Matrix_Unique_ID or MUI.

The ID portion of the parameters has two parts. ID=11738756:0

11738756 is the Matrix_Unique_ID of the listing, our primary key, as noted in the METADATA-RESOURCE. It is not the MLS number. This has caught other users in the past.

The “:0” of the ID means to get the primary picture. A “:1” will get you the first picture, which on our system is the primary picture, so in effect :0 and :1 are equivalent. “:2” will give you the second picture. If you use a “:*” you will receive a multipart response which contains all of the images for that property.

There is no request in RETS to simply ask for ALL of the images.

There is a field called PhotoCount in the listing tables that indicates the number of photos for that particular listing.

We have added a new field to the listings called PhotoModificationTimestamp. This field contains a modification timestamp for the photos for a listing. So if any new pictures are added, removed or replaced the listing’s PhotoModificationTimestamp would be changed as well as the MatrixModifiedDT.

You can pull photo updates for listings using a similar method for pulling listing updates, but instead of using MatrixModifiedDT you can use PhotoModificationTimestamp. When listings have a newer PhotoModificationTimestamp, you need to download the photos for that listing. Alternatively, you could use a different method whereby when you download updates for a listing, you can compare the old PhotoModificationTimestamp with the new one. If they differ, re-download the photos for that listing. This method works because the MatrixModifiedDT is updated at the same time as the PhotoModificationTimestamp.