

MLS Listings API

MLS Listings API provides listings data using **RESO** (Real Estate Standards Organization) namespace convention. The listings data are downloaded from MLS Sources and processed by EAP MLS Data Pipeline (MDP) which enriches raw Listings, Images and OpenHomes data at regular intervals.

As the MLS Sources gets on-boarded in the MLS Data Pipeline, the MLS Source are mandatorily enabled to accessible for the users. You must use MLS Display Rules API to get the status of MLS sources to receive listings.

MLS Listings APIs provide various endpoints to get listings data. For more details, refer <https://developers.realogy.com/api-product/mls-data-service/specification?apiname=mls>.

Sample use cases:

Get MLS Listings by Listing Id

A listing is associated to more than one MLS source. The response has more than one listing for a given listing id.

<https://api.realogy.com/v1.0/mls/listing/12345>

Get MLS Listings by Listing Id and MLS Source

This API returns exactly one listing for a specified MLS source.

https://api.realogy.com/v1.0/mls/listing/12345/source/AR_TMLS

Get MLS Listings by MLS Source

This API returns listings associated to a specific MLS source.

https://api.realogy.com/v1.0/mls/source/AR_TMLS

How to paginate the listings response?

By default, "*MLS Listings by Source*" API returns maximum of 20 listings in the response. Pagination (offset & limit) are used to receive more data. The `offset` is the starting index to fetch MLS listings. The `limit` is used to specify the total number of listings to be sent in the response.

https://api.realogy.com/v1.0/mls/source/AR_TMLS?offset=0&limit=20

How to get active listings?

By default, "*MLS Listings by Source*" API returns both "ACTIVE" and "INACTIVE" listings in the response. The filter query parameter (*standardStatus*) is used to receive listings for a specific listing status.

For example, the following API call returns only first 20 "ACTIVE" listings.

https://api.realogy.com/v1.0/mls/source/AR_TMLS?offset=0&limit=20&filter.standard

Status=ACTIVE

How to get listings delta based on the listings change date?

"MLS Listings by Source" API listings accepts "*lastChangeTimestamp*" as query parameter. This timestamp is used to get listings based on the last update on date and/or time of a listings. The internal Data pipeline updates the changed date of a listing when there is an update to a listing from the MLS Sources.

https://api.realty.com/v1.0/mls/source/OH_CBR?offset=0&limit=20&lastChangeTimestamp=2020-03-01T00:00:00.000Z

The request must return all the listings that are updated after the given timestamp. You must provide the timestamp in the UTC format as shown on the above given example. By default, the API ignores milli and nano seconds from the request to search the listings but includes seconds.

Note: Refer the swagger definition for latest filters.

MLS Changes API

This API is used to receive listings changes in real time or sequentially in the order of changes with `http2` streaming.

For example, the following API call receives all the listings changes irrespective of MLS sources:

<https://api.realty.com/v1.0/mls/changes>

Response:

The response to Changes API request is in the following format:

```
{
  "result": {
    "mlsChange": {
      "marker": "<UNIQUE_ID_OF_THIS_CHANGE>",
      "changeType": "<insert/update/replace>",
      "changeTime": "<TIME_THE_CHANGE_HAPPENED>"
    },
    "mlsListing": {ACTUAL_PAYLOAD_OF_LISTINGS},
    "mlsId": "<UNIQUE_ID_OF_LISTINGS>"
  }
}
```

- To receive listings changes to a specific MLS source, the "*sourceSystemKey*" parameter is used.
For example,
https://api.realty.com/v1.0/mls/changes?sourceSystemKey=AR_TMLS.
- By default, **ChangesAPI** returns listings changes related to the change types as follows:
 - Insert
 - Update
 - Replace

To receive changes related to a specific change type, ***changeType*** query parameter is used in the request,

https://api.realty.com/v1.0/mls/changes?sourceSystemKey=AR_TMLS&changeType=insert.

- Using **marker**:

The API client is expected to store the last successful "**marker**" from the response to resume the changes during the next API invocation. For example, let's assume the last successful response is,

```
{
  "result": {
    "mlsChange": {
      "marker":
      "825DFB3330000000012B022C0100296E5A1004298DDC58EEA94745877D819EC2D2EB05
      463C5F6964003C41525F544D4C535F313030303631000004",
      "changeType": "replace",
      "changeTime": "2019-12-19T08:22:08.000000001Z"
    },
    "mlsListing": {
      "property": {
        "propertyType": "LAND", .....(skipped the
        response intentionally)
      },
      "mlsId": "AR_TMLS_100061"
    }
  }
}
```

To resume the listings changes using **marker**,

https://api.realogy.com/v1.0/mls/changes?sourceSystemKey=AR_TMLS&marker=825DFB3330000000012B022C0100296E5A1004298DDC58EEA94745877D819EC2D2EB05463C5F6964003C41525F544D4C535F313030303631000004

"**marker**" is also used to handle failure cases to resume the changes from last successful change.

Timeout:

ChangesAPI request is timed out if the connection is idle for 60-120 secs without receiving listings changes. This can happen when there are no changes to the listings that needs to send to the API client. The API client is expected to handle this timeout error (**Http 504**) and resume the events by using the "marker" as explained above.

Get MLS Display Rules by Source

This API is used to check if a given MLS source is "**Active**" and able to get listings. Returns no data if a given MLS Source is "Inactive" or does not exist.

https://api.realogy.com/v1.0/displayrules/mls/displayrules/source/AR_TMLS

Alternatively, the following API is used to get list of all active MLS Sources,

<https://api.realogy.com/v1.0/displayrules>.

You can refer to the Realogy API Portal (Master Data Services API Product) for more details on the API Definitions.

Glossary:

- **Listing Id** (*listingId*): A unique id for a listing.
- **MLS Sources** (*sourceSytemKey*): A unique string id to represent MLS source.
- **Marker** (*marker*): A unique id returned in the response of Changes API.
- **Standard Status** (*standardStatus*): Status of a listing. It is either "ACTIVE" or "INACTIVE".

FAQ's:

[What are the fields that corresponds to Rental and Commercial Listings?](#)

Rental and Commercial Fields:

property.financial.incomeIncludes

property.financial.totalActualRent

Rental Fields:

property.financial.rentIncludes
property.financial.isRentControl
property.listing.price.petRent
property.listing.price.monthsRentUpfront
property.characteristics.leaseTerm
property.characteristics.isRentersInsuranceRequired
property.business.leaseAmountFrequency

Commercial Fields:

property.financial.salesIncludes
property.financial.electricExpense
property.financial.tenantPays
property.financial.ownerPays
property.listing.buildingPermits
property.listing.documentsAvailable
property.listing.disclosures
property.structure.approximateOfficeSquarefeet
property.structure.approximateRetailSquarefeet
property.structure.approximateWarehouseSquarefeet
property.structure.totalRestrooms
property.structure.load
property.utilities.numberOfSeparateElectricmeters
property.utilities.numberOfSeparateGasmeters
property.utilities.numberOfSeparateWatermeters

[What are the limitations of ChangesAPI?](#)

- Request to ChangesAPI sends the listings changes in real time. It is possible that the request gets time out (**Http 504**) after 60-120 seconds when there are no changes to the listings happened.
- API client is expected to handle the failure and re-issues the request after a certain interval.
- ChangesAPI responds listings with "**marker**". You can store the **marker** value on the API client side and use them to resume listings changes.

[What is the list of possible Property Type?](#)

The possible property types are SFR (Single Family Residence), MFR (Multi-Family Residence), MFD (Manufactured/Mobile Homes), condo, townhouse, coop, farm, land, rental, Commercial_sale, Commercial_lease, and unknown.

[What is the list of possible Standard Status?](#)

The status of the listing as it reflects the state of the contract between the listing agents and seller or an agreement with a buyer are active, inactive, sold, cancelled, hold, unknown, expired, temp, terminated, pending, and withdrawn.

Are the string fields being in Uppercase?

MLS Data Pipeline, as part of enrichment phase capitalize "*name*" fields into uppercase as of now. The letter case is not guaranteed for remaining MLS fields as it varies between MLS Sources.

Are the ID fields in the listings are unique?

ID fields are unique per MLS Source. It is not guaranteed to be unique across all the MLS Sources.

What is the unit for "*buildingAreaTotal*"?

The unit for BuildingAreaTotal is Square Feet.

What is *MasterId*?

MasterIds is an Id generated that are unique across multiple listings and sources.

For example: "*addressMasterId*" is same for an address across multiple listings and MLS Sources.

If the MasterId is not available, we will notify when this feature is enabled and MasterId is available.

What is the list of *MasterId*'s?

Please refer the API Documentation for the latest details of MasterId list. The present list of MasterIds are as follows:

- ListingMasterId
- PropertyMasterId
- ListAgentMasterId
- ListOfficeMasterId
- CoListAgentMasterId
- CoListOfficeMasterId
- BuyerAgentMasterId
- BuyerOfficeMasterId
- CoBuyerAgentMasterId
- CoBuyerOfficeMasterId
- AddressMasterId

How can I get listings statistics?

MLS Listings Statistics API is used to get listings statistics such as "Total Listings", "Total Listings with Photos", "Total Listings with OpenHomes", "Total Listings by Property Type" and so on. Please refer the MLSListingsStats API documentation for more details.

What is the use of MLS Display Rules API?

The MLS Display Rules API is to check if a given MLS Source is active in the EAP MLS platform to receive listings data.