
# MRIS Data Services

## Getting Started Guide

**Document Version 1.0**

**April 14, 2011**

# Table of Contents

# SECTION 1 INTRODUCTION

## 1.1 Purpose of this Document

This document describes the fundamentals of MRIS Data Services (MDS) and tells what's needed to get started using them.

## 1.2 Audience

This document is intended for use by developers and business analysts.

## 1.3 What are MRIS Data Services (MDS)?

MRIS Data Services provide real estate information in an easy to use manner. The specification is implemented as RESTful web services. The data is encapsulated in the form of Resources that contain the consolidated information about a given real estate entity that a user might be interested in. The web services are themselves accessible over HTTP, and all it takes is an HTTP client to make the calls and get the data.

Prior to the RESTful services, the data was only accessible through a RETS (Real Estate Transaction Standard) interface. That interface provides a standard, defined by the Nationals Association of Realtors (NAR), to exchange data in a uniform manner among the various Multiple Listing Agencies (MLS) and other customers.

The RESTful web services provide an open set of APIs any web developer can use without having to understand the complexities of the real estate data or real estate standards. This also helps developers that are not in the real estate industry.

## 1.4 What is REST?

Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems. As such, it is not only a method of building what are sometimes called 'web services.' REST strictly refers to a collection of architecture principles that outline how resources are defined and addressed. Systems that follow REST design principles are often referred to as 'RESTful.'

The term Representational State Transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation. (Fielding, 2000) Fielding is one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification versions 1.0 and 1.1.

# SECTION 2 MDS SERVICES

## 2.1 Getting Started

### 2.1.1 What is required?

To access MDS Services you need to know the following information:

- User-id
- Password
- User-Agent
- Service Endpoint, the entry point to a service

### 2.1.2 How do I request MDS Services?

All MDS Services are requested through HTTP 1.1 requests. HTTP GET requests are used to retrieve resources.

### 2.1.3 What services are available?

A full range of Create, Read, Update and Delete (CRUD) transactions are available. This document tells how to request (read) resources.

### 2.1.4 What resources are available?

You can retrieve a complete list of available resources from the Resource Metadata.

## 2.2 Format of Messages

### 2.2.1 Required Headers

#### User-Agent

The User-Agent header contains information about the user-agent originating the request. This information is used to compile statistics, trace protocol violations, and automate recognition of user-agents for the sake of tailoring responses to avoid particular user-agent limitations. It also provides enhanced capabilities to some user-agents.

All client requests MUST include this field. This is a standard HTTP header field as defined in RFC 2616.

#### Cookie

The client MUST implement cookie handling as specified in RFC 2109. If any server response has included a valid `Set-Cookie` header, and the cookie in that header has not expired, the client MUST return the corresponding `Cookie` header. See RFC 2109 for the full specification. Cookies may be used to manage a client's session state.

#### Accept

The Accept request header field can be used to specify the media types that are acceptable for the response. Accept headers can be used to indicate that the request is specifically limited to a small set of desired types, as in the case of a request for an inline image.

#### Authorization

The client MUST implement authentication handling as specified in RFC 2617; see section 4.2, "Authentication of Clients using Digest Authentication" for more information. If any server response has included a `WWW-Authenticate` header, the client MUST respond with the corresponding Authorization header.

### 2.2.2 Authentication

Clients need to be authenticated (to establish their identity) before they can access any data through the MRIS Data Services. HTTP provides a simple challenge-response authentication mechanism that may be used by a server to challenge a client request, and by a client to provide authentication information. An origin server uses the 401 (Unauthorized) response messages to challenge the authorization of a user-agent.

A digest authentication mechanism is used to authenticate the clients. This mechanism is specified by RFC 2617 of the HTTP protocol. This is an extension from the basic authentication, where the user-id and password are sent over the network in clear text. In digest authentication, the clients send a digest of the credentials to the server. Digest authentication is basically an application of MD5 cryptographic hashing with usage of nonce values to prevent cryptanalysis.

### 2.2.3 Addressing Resources

All MDS Services are addressed through a Uniform Resource Identifier (URI) that identifies the requested resource.

#### Parts of the URI

#### Service Endpoint

The URIs for all MDS services begin with the MDS Service Endpoint. A service endpoint is the entry point to a service; it is where the service connects to the network. A client can consider a service endpoint as the interface to the service. The endpoint makes it possible for a client to access any services and/or resources by appending the path of the service or resource to the service endpoint. URLs of service endpoints are system-defined.

#### Resource Name

Following the Service Endpoint, you specify the requested resource.

You can retrieve a list of available resources from the Resource Metadata.

If you access a resource directly without providing a unique key, the server will return a collection of resources. A collection will include only the keys to the individual resources. (For large collections, the number of records returned may be limited depending on the authenticated user's access roles.)

Example:

```
http://data.mris.com/services/rest/Property
```

### Resource Keys

Accessing a resource within the resource collection is done by specifying the resource key in the sub-path. You may also request multiple keys, by separating them with a comma.

Example:

```
http://data.mris.com/services/rest/Property/1234567,2345678
```

### Query Strings

You may specify additional options in the query string:

- Count – Return a count of records instead of a list of keys.

  Example:
  `http://data.mris.com/services/rest/Property?Count=1`

- Limit – Restrict the number of keys returned.

  Example:
  `http://data.mris.com/services/rest/Office?Limit=10`

- Query – Filter which resource keys are returned.

  Example:
  `http://data.mris.com/services/rest/Property?Query=(ListPrice GTE 375000)`

- View – Specify a view of the data to be returned.

  Example:
  `http://data.mris.com/services/rest/Agent/1234567890?View=AGENT_PARTY_ROLES`

- Select – Specify fields to be returned.

  Example:
  `http://data.mris.com/services/rest/Property/1234567890?Select=ListingID,ListPrice,ListingStatus`

## 2.3 Formats and Metadata

### 2.3.1 Schema

A schema formally describes what a given XML document contains, in the same way a database schema describes the data that can be contained in a database (table structure, data types, etc.). An XML schema describes the coarse shape of the XML document, such as what fields an element can contain, which sub elements it can contain, and so on. It can also describe the values that can be placed into any element or attribute. All MDS return messages share the same schema.

### 2.3.2 Metadata

Metadata enables a client to better format the data for display, provide an intuitive search interface, and properly handle the creation or updates to a resource.

Metadata is organized by Resource which describes all of the data within a resource collection. This allows a client to effectively use the metadata for display, search, and update purposes.

All metadata is retrieved by issuing an HTTP GET transaction to a URI, indicating which metadata is requested.

Common metadata is described by, but not limited to, the following table:

| | Path | Description |
|---|---|---|
| Resource | /Resource | Provides information about the resources available in the system |
| Field | /Metadata/<ResourceName>/Resource | Provides information about fields within a resource |
| Lookup | /Metadata/<ResourceName>/Lookup | Provides information about lookups that are available to fields as single- and multi-valued lookups. |
| Lookup Item | /Metadata/<ResourceName>/Lookup/<LookupName> | Provides information about lookup items that belong to a particular lookup. |

## 2.4 UserSessionInfo Service

Information about the user's current session can be retrieved by using the UserSessionInfo service.

**Request Path:** `/UserSessionInfo`

**Example:** http://data.mris.com/services/UserSessionInfo

## 2.5 Logout Service

Logout can be requested by using the Logout service. It is recommended that all clients enforce logout. A logout response will return the Response payload with the Status of the logout. The Status node tells whether the logout was successful or an error or any other condition occurred.

**Request Path:** `/Logout`

**Example:** http://data.mris.com/services/Logout

# SECTION 3 REFERENCES

## 3.1 Specifications and Documentation

### 3.1.1 Hypertext Transfer Protocol: HTTP 1.1 (RFC 2616)

http://tools.ietf.org/html/rfc2616

### 3.1.2 HTTP Authentication: Digest Access Authentication (RFC 2617)

http://tools.ietf.org/html/rfc2617

### 3.1.3 Cookies (RFC 2109)

http://tools.ietf.org/html/rfc2109

## 3.2 REST References

### 3.2.1 RESTful definition

http://en.wikipedia.org/wiki/Restful

### 3.2.2 RESTful Java with JAX-RS, O'Reilly

## 3.3 Works Cited

Fielding, R. (2000). *Fielding Dissertation: Chapter 5: Representational State Transfer (REST)*. Retrieved from http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm