

# API03 Search Syntax

## General Search Syntax

`http://127.0.0.1:9000/nmc/rpc/search?server=[BOOKMARK]search=[hex-encoded query]start=[value]count=[value]`

Performs a search for the specified query. The search parameter is better described below.

Note: The search query must be hex-encoded. For paging through a search, the start and count parameters can be used.

The search syntax takes two forms.

One can use the UPnP search syntax:

`upnp:class derivedfrom "object.item.audioItem.musicTrack" and (dc:title contains "FindThis" or upnp:genre contains "FindThis" or upnp:artist contains "FindThis")`

Once hex-encoded, this can be used with a search RPC call as follows:

```
http://127.0.0.1:9000/nmc/rpc/search?server=uuid%3A4f1bf61c-599b-443d-b41d-36408362ec75,
0&search=75706e703a636c617373206465726976656466726f6d20226f626a6563742e6974656d2e617564696f4974656d2e6d75736963547
261636b2220616e64202864633a7469746c6520636f6e7461696e73202246696e645468697322206f722075706e703a67656e726520636f6e7
461696e73202246696e645468697322206f722075706e703a61727469737420636f6e7461696e73202246696e64546869732229&start=0&count=
10
```

One can also use a simplified search syntax:

`type=musicItem&artist=FindThis&album=FindThat`

This also must be hex-encoded if used in the RPC, and would look like:

```
http://127.0.0.1:9000/nmc/rpc/search?server=uuid%3A4f1bf61c-599b-443d-b41d-36408362ec75,
0&search=747970653d6d757369634974656d266172746973743d46696e645468697326616c62756d3d46696e6454686174&start=0&count=
10
```

## Simplified Search Syntax

The simpler search syntax is presented below and uses URL-encoded parameters to set up the search. Caveat: These search tokens are combined into an AND search, e.g. a searched-for item must have all of the token values in its metadata before it matches. There is currently no provision for OR searches.

By default, these searches will use the 'contains' search semantic, meaning that they will match substrings in the metadata. e.g. 'title=FindThis' will match when a title is 'FindThisThing'.

To force an exact, whole-word match of a metadata string, append the 'exact=1' parameter somewhere in the search string. e.g. 'title=FindThis&exact=1' will not match when a title is 'FindThisThing'.

It will only match when a title is 'FindThis'.

The following parameters are allowed in a **URL-encoded** query string format.

- keyword - value
- title - the title
- creator - the creator
- genre - the genre
- album - the album
- description - description
- date - date
- class - media item's class
- author - the author
- actor - the actor
- director - director
- albumArtist - album artist
- trackNumber - track number
- seriesTitle - series title
- seriesID - series ID
- episodeNumber - episode number
- episodeCount - episode count
- channelNr - channel number
- channelName - channel name
- longDescription - long description
- rating - upnp:rating
- resolution - resolution
- duration - duration
- id - @id
- refID - @refID
- protocolInfo - protocol info
- userRating - pv:rating

type - the type

The predefined "type" values are:

musicItem, musicAlbum, musicArtist, musicGenre,  
photoItem, photoAlbum,  
videoItem,  
playlist,  
folder,  
container,  
item

## Examples

Find all music tracks that contain FindThis in the title.

```
type=musicItem&title=FindThis
```

Find all music tracks in the FindThis album.

```
type=musicItem&album=FindThis
```

Find all music tracks with FindThis as the artist, and FindThat as the album.

```
type=musicItem&artist=FindThis&album=FindThat
```

Find all music albums with FindThis as the genre.

```
type=musicAlbum&genre=FindThis
```

URL encoding of special characters searching for music item.

Actual title: a"&=b

URL encoded string: title=a%22%26%3db&type=musicItem

## UPnP Search Syntax

The UPnP Search Syntax grammar is as follows:

```
searchCrit ::= searchExp | asterisk
searchExp ::= relExp | searchExp wChar+ logOp wChar+ searchExp | '(' wChar* searchExp wChar* ')'
logOp ::= 'and' | 'or'
relExp ::= property wChar+ binOp wChar+ quotedVal | property wChar+ existsOp wChar+ boolVal
binOp ::= relOp | stringOp
relOp ::= '=' | '!=' | '<' | '<=' | '>' | '>='
stringOp ::= 'contains' | 'doesNotContain' | 'derivedfrom'
existsOp ::= 'exists'
boolVal ::= 'true' | 'false'
quotedVal ::= dQuote escapedQuote dQuote
wChar ::= space | hTab
```

## Examples

Find all music tracks that contain FindThis in the title.

```
upnp:class derivedfrom "object.item.audioItem.musicTrack" and (dc:title contains "FindThis")
```

Find all music tracks in the FindThis album.

```
upnp:class derivedfrom "object.item.audioItem.musicTrack" and (upnp:album contains "FindThis")
```

Find all music tracks with FindThis as the artist, and FindThat as the album.

```
upnp:class derivedfrom "object.item.audioItem.musicTrack" and (upnp:artist contains "FindThis")
```

Find all music albums with FindThis as the genre.

```
upnp:class derivedfrom "object.item.audioItem.musicTrack" and (upnp:genre contains "FindThis")
```

## Exact Searches

Find a music track with "FindThis" as the title.

```
upnp:class derivedfrom "object.item.audioItem.musicTrack" and (dc:title = "FindThis")
```

## Searchable Fields

Items can be searched for by specifying some combination of the following UPnP fields.

```
res@resolution
res@duration
dc:title
dc:creator
upnp:actor
upnp:artist
upnp:genre
upnp:album
dc:date
upnp:class
@id
@refID
@protocolInfo
upnp:author
dc:description
pv:avKeywords
pv:rating
upnp:seriesTitle
upnp:episodeNumber
upnp:director
upnp:rating
upnp:channelNr
upnp:channelName
upnp:longDescription
pv:capturedate
pv:custom
```

## derivedfrom

derivedfrom can also be one of the following:

```
object.item
object.item.audioItem.musicTrack
object.item.audioItem.audioBroadcast
object.item.audioItem.online.musicTrack
object.item.imageItem.photo
object.item.imageItem.online.photo
object.item.videoItem.movie
object.item.videoItem.classified.movie
object.item.videoItem.videoBroadcast
object.item.videoItem.online.movie
object.container
object.container.playlistContainer
object.container.musicContainer
object.container.pictureContainer
object.container.videoContainer
object.container.storageFolder
object.container.album.musicAlbum
object.container.album.photoAlbum
object.container.person.musicArtist
object.container.genre.musicGenre
```