



# Extraction API

## *Version 3*

2017-05-10

### Overview:

We offer the ability to export all database contents, with few exceptions. We provide result pagination to improve response times and response size. To ensure reliability and stability across all our clients we implement a quota system. To enforce quota limits, requests must be properly authorized via token based authorization.

### Intended Uses:

We implement RESTful services of GET and POST methods over https. Authentication and Authorization are supported by the "Authorization: Bearer {{token}}" header HTTP construct. Security is based on a pre-shared token system, where we manage the tokens and provide them to our clients for later use. Load-inducing queries are deducted from quota. These include certain detailed help messages, and search/export queries.

Export requests may include additional parameters to reduce locations and customize the level of detail returned for each matching location. These parameters have been modelled after the SQL constructs of *select*, *from* (for multi-domain clients), and *where*. We support ordering, limits, and paging through additional parameters.

Clients may issue POST commands with all parameters in the POST body as properly formatted JSON, or they may provide certain override conditions in the URL to support non-filtering queries. URL-overriding parameters include: *orderBy*, *sortDesc*, and *limit*. Paging is built into the URL path. These features allow a developer to conveniently test and debug queries without providing a POST body.

## HTTP Response Codes & Descriptions

- 200 Success
- 400 Bad Request // *supplemental info may be provided*
- 401 Unauthorized, check your token.
- 401 Unauthorized, you must specify a token.
- 404 Not found // *reason may be provided*
- 404 Not found, check the url. // *bad route*
- 405 Method not allowed // *list of allowed HTTP methods provided*
- 429 Too many requests, quota exceeded.
- 500 Internal Server Error // *exception data may be provided*
- 501 Not implemented // *hint provided*

## API Tools

- 1) Curl
- 2) Wget
- 3) Postman
- 4) Browser Plugins (eg. HttpRequester)

## Quick Start Guide

- 1) Request an API token.
- 2) View the most up-to-date help message.
  - a. GET <https://search.api.rioseo.com/v3/help>  
Authorization: Bearer {{token}}
- 3) List capabilities for your domain.
  - a. GET <https://search.api.rioseo.com/v3/fields/select>  
Authorization: Bearer {{token}}
  - b. GET <https://search.api.rioseo.com/v3/fields/from>  
Authorization: Bearer {{token}}
  - c. GET <https://search.api.rioseo.com/v3/fields/where>  
Authorization: Bearer {{token}}
- 4) Start scripting.
  - a. POST <https://search.api.rioseo.com/v3/search>  
Authorization: Bearer {{token}}

## Home – GET

This API call will show all supported methods for usage.

### API Request:

```
curl -X GET "https://search.api.rioseo.com"

GET https://search.api.rioseo.com
```

### API Response:

```
{
  "supported methods": [
    "help",
    "search",
    "fields/select",
    "fields/from",
    "fields/where"
  ],
  "version": "3"
}
```

## Help – GET

This API call will show you the help message.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/help" -H "Authorization:
Bearer {{token}}"

GET https://search.api.rioseo.com/help
Authorization: Bearer {{token}}
```

### API Response:

```
{
  "routes": {
    "/v3/help": "This help message.",
    "/v3/search": "Supports GET and POST with key=value. Searches for...",
    "/v3/search/{page}": "Paged search results.",
    "/v3/search/{page}/{limit}": "Paged search results with custom max...",
    "/v3/fields/select": "Request listing of exportable fields.",
    "/v3/fields/from": "Request listing of domains.",
    "/v3/fields/where": "Request listing of searchable fields."
  },
  "params": {
    "help": null,
    "search": {
      "select": {
        "types": [
          "string",
          "array"
        ],
        "methods": [
          "POST"
        ],
        "info": {
```

```

        "string": "JSON-wrapped comma separated list of data...",
        "array": "JSON-wrapped array of data fields to return..."
    },
    "default": "results include a subset of available fields",
    "examples": {
        "as-string": {
            "select": "fid,address_1,city,region,post_code"
        },
        "as-array": {
            "select": [
                "fid",
                "address_1",
                "city",
                "region",
                "post_code"
            ]
        }
    },
    "IMPORTANT": "JSON requires strings be enclosed in double..."
},
"from": {
    "type": "string",
    "methods": [
        "GET",
        "POST"
    ],
    "info": "Domain name of export being requested.",
    "default": "first available",
    "examples": {
        "as-GET": "from=example.com",
        "as-POST": {
            "from": "example.com"
        }
    }
},
"where": {
    "type": "application/json",
    "methods": [
        "POST"
    ],
    "info": "Properly formatted JSON body wrapped in a 'where'...",
    "default": "empty. Does not filter. Returns all records in...",
    "example": {
        "where": {
            "zip": "90210",
            "boolean.open_saturdays": 1
        }
    },
    "IMPORTANT": "JSON requires strings be enclosed in double-..."
},
"orderBy": {
    "type": "string",
    "methods": [
        "GET",
        "POST"
    ],
    "info": "Name of response field to sort by",

```

```

        "default": "orders by internal document id",
        "IMPORTANT": "Avoid using orderBy if you will be iterating..."
    },
    "sortDesc": {
        "type": "bool",
        "methods": [
            "GET",
            "POST"
        ],
        "info": "If set and not explicitly 'false' or '0', sorts...",
        "default": "false. default sort order is ascending."
    },
    "{page}": {
        "type": "integer",
        "info": "Page index of chunked results",
        "default": 1
    },
    "{limit}": {
        "type": "integer",
        "info": "Number of records per page",
        "default": 100
    }
},
"fields": {
    "select": null,
    "from": null,
    "where": null
}
},
"version": "3",
"quota": {
    "remaining": 151,
    "limit": 214
}
}

```

### Potential Error:

If help request is unauthenticated, you will get an error.

#### API Request:

```
curl -X GET "https://search.api.rioseo.com/help"
```

```
GET https://search.api.rioseo.com/help
```

#### API Response:

```

{
  "error": {
    "code": 401,
    "description": "Unauthorized, you must specify a token.",
    "url": "/v3/help"
  }
}

```

## Select – GET

This API call shows a list of all selectable elements. These results will be used in the “select” parameter.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/fields/select" -H
"Authorization: Bearer {{token}}"

GET https://search.api.rioseo.com/fields/select
Authorization: Bearer {{token}}
```

### API Response:

```
[
  "lid",
  "address_1",
  "address_2",
  "suite",
  "city",
  "country",
  "fid",
  "from_email",
  "lat",
  "lng",
  "post_code",
  "post_code_plus",
  "region",
  "website",
  "profile_fields.1.profile_indy_0_0_location_description",
  "profile_fields.1.profile_indy_2_0_page_title",
  "hours_sets.primary",
  "hours_sets.primary.days.monday",
  "hours_sets.primary.days.monday.open",
  "hours_sets.primary.days.monday.close",
  "hours_sets.primary.days.tuesday",
  "hours_sets.primary.days.tuesday.open",
  "hours_sets.primary.days.tuesday.close",
  "hours_sets.primary.days.wednesday",
  "hours_sets.primary.days.wednesday.open",
  "hours_sets.primary.days.wednesday.close",
  "hours_sets.primary.days.thursday",
  "hours_sets.primary.days.thursday.open",
  "hours_sets.primary.days.thursday.close",
  "hours_sets.primary.days.friday",
  "hours_sets.primary.days.friday.open",
  "hours_sets.primary.days.friday.close",
  "specialties_object.locator.name",
  "specialties_object.standard_mobile_page"]
```

## Select – POST

This API call performs a search, specifying the fields to return in the “select” parameter.

### API Request:

```
curl -X POST "https://search.api.rioseo.com/search/2"
-H "Authorization: Bearer {{token}}" -H "Content-Type:
application/json"
-d
'{"select": "fid, address_1, city, region, profile_fields.1.profile_indy_0_0
_location_description, hours_sets.primary"}'

POST https://search.api.rioseo.com/search
Authorization: Bearer {{token}}
Content-Type: application/json
{"select": "fid, address_1, city, region, profile_fields.1.profile_indy_0_0
_location_description, hours_sets.primary"}
```

### API Response:

```
{
  "records": [
    {
      "address_1": "W 708 Boone",
      "city": "Spokane",
      "fid": "1014",
      "region": "WA",
      "profile_fields": {
        "1": {
          "Profile:Indy:0_0_location_description": "Starbucks locations serve
hot and cold beverages, whole-bean coffee, microground instant coffee, full-
leaf teas, pastries, and snacks. Most stores also sell pre-packaged food
items, hot and cold sandwiches, and items such as mugs and tumblers.
Starbucks Evenings locations also offer a variety of beers, wines, and
appetizers after 4 p.m.[6] Through the Starbucks Entertainment division and
Hear Music brand, the company also markets books, music, and film. Some of
the company's products are seasonal or specific to the locality of the store.
Starbucks-brand ice cream and coffee are also offered at grocery stores."
        }
      },
      "hours_sets": {
        "primary": {
          "label": "Primary Hours",
          "name": "primary",
          "type": "0",
          "days": {
            "Sunday": [
              {
                "open": "11:00",
                "close": "15:00"
              }
            ],
            "Monday": [
              {
                "open": "09:00",
                "close": "17:00"
              }
            ]
          }
        }
      }
    }
  ]
}
```

```
    ],
    "Tuesday": [
      {
        "open": "09:00",
        "close": "17:00"
      }
    ],
    "Wednesday": [
      {
        "open": "09:00",
        "close": "17:00"
      }
    ],
    "Thursday": [
      {
        "open": "09:00",
        "close": "17:00"
      }
    ],
    "Friday": [
      {
        "open": "09:00",
        "close": "16:00"
      }
    ],
    "Saturday": [
      {
        "open": "08:00",
        "close": "20:00"
      }
    ]
  ]
}
}
}
},
"metadata": {
  "page": {
    "current": 1,
    "last": 143
  },
  "index": {
    "from": 1,
    "to": 1,
    "of": 143
  },
  "version": "3",
  "quota": {
    "remaining": 30,
    "limit": 50
  }
}
}
```

## From – GET

This API call shows a list of all domains authorized for the given token. These results will be used in the “from” parameter.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/fields/from" -H
"Authorization: Bearer {{token}}"

GET https://search.api.rioseo.com/fields/from
Authorization: Bearer {{token}}
```

### API Response:

```
{
  "444": "rioseo.com"
}
```

## From – POST

This API call performs a search, returning the default fields and specifying the domain to use in the “from” parameter.

### API Request:

```
curl -X POST "https://search.api.rioseo.com/search/2"
-H "Authorization: Bearer {{token}}" -H "Content-Type:
application/json"
-d '{"from": "446"}'

POST https://search.api.rioseo.com/search
Authorization: Bearer {{token}}
Content-Type: application/json
{"from": "446"}
```

### API Response:

```
{
  "records": [
    {
      "address_1": "W 708 Boone",
      "address_2": null,
      "suite": null,
      "city": "Spokane",
      "fid": "1014",
      "lat": null,
      "lng": null,
      "location_name": "Spokane",
      "post_code": "99201",
      "region": "WA"
    }
  ]
}
```

## Where – GET

This API call shows a list of all filterable elements. These results will be used in the “where” parameter.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/fields/where" -H
"Authorization: Bearer {{token}}"

GET https://search.api.rioseo.com/fields/where
Authorization: Bearer {{token}}
```

### API Response:

```
[
  "lid",
  "address_1",
  "address_2",
  "suite",
  "city",
  "country",
  "fid",
  "from_email",
  "lat",
  "lng",
  "post_code",
  "post_code_plus",
  "region",
  "website",
  "profile_fields.1.profile_indy_0_0_location_description",
  "profile_fields.1.profile_indy_2_0_page_title",
  "boolean.locator.store",
  "boolean.locator.outet",
  "boolean.standard_mobile_page.view_mobile_page"]
```

## Where – POST

This API call performs a search, returning the default fields and specifying the fields to return in the “where” parameter.

### API Request:

```
curl -X POST "https://search.api.rioseo.com/search/2"
-H "Authorization: Bearer {{token}}" -H "Content-Type:
application/json"
-d '{"where":{"city":"San Diego"}}'

POST https://search.api.rioseo.com/search
Authorization: Bearer {{token}}
Content-Type: application/json
{"where":{"city":"San Diego"}}
```

### API Response:

```
{
  "records": [
    {
      "address_1": "9225 Clairemont Mesa",
      "address_2": null,
      "suite": null,
```

```

    "city": "San Diego",
    "fid": "776",
    "lat": null,
    "lng": null,
    "location_name": "Clairemont Mesa",
    "post_code": null,
    "region": "AL"
  }
],
"metadata": {
  "page": {
    "current": 1,
    "last": 1
  },
  "index": {
    "from": 1,
    "to": 1,
    "of": 1
  },
  "version": "3",
  "quota": {
    "remaining": 23,
    "limit": 50
  }
}
}
}

```

## Search – GET

This API call performs a basic search using the default “select” parameters and no filters.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/search" -H "Authorization: Bearer {{token}}"
```

```
GET https://search.api.rioseo.com/search
Authorization: Bearer {{token}}
```

### API Response:

```

{
  "records": [{
    "address_1": "9255 Towne Center Dr",
    "address_2": null,
    "suite": "750",
    "city": "San Diego",
    "fid": "1",
    "lat": "32.8760187",
    "lng": "-117.2087176",
    "location_name": "Rio SEO",
    "post_code": "92121",
    "region": "CA"
  }],
  "metadata": {
    "page": {
      "current": 1,
      "last": 1
    }
  }
}

```

```
    },
    "index": {
      "from": 1,
      "to": 1,
      "of": 1
    },
    "version": "3",
    "quota": {
      "remaining": 45,
      "limit": 50
    }
  }
}
```

## Page 2

This API call will show the second page of search results.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/search/2" -H "Authorization: Bearer {{token}}"
```

```
GET https://search.api.rioseo.com/search/2
Authorization: Bearer {{token}}
```

### API Response:

[ Requests second page of results, if more than one page available, else 404 page not found. ]

## Page 2, limit 15

This API call will show the second page of search results, with 15 results per page, returning locations 16-30.

### API Request:

```
curl -X GET "https://search.api.rioseo.com/search/2/15" -H "Authorization: Bearer {{token}}"
```

```
GET https://search.api.rioseo.com/search/2/15
Authorization: Bearer {{token}}
```

### API Response:

[ Requests second page of 15 locations. Will return locations 16-30. Use *orderBy* to specify sort field. ]

## Search – POST

Important tip! The POST method expects properly formatted JSON. Use double-quotes around object names and strings. Single quotes are not valid JSON.

This API call performs a filtered search, specifying the fields, returning the second page of results.

### API Request:

```
curl -X POST "https://search.api.rioseo.com/search/2"
-H "Authorization: Bearer {{token}}" -H "Content-Type:
application/json"
-d
'{"select":"address_1,address_2,suite,city,fid,lat,lng,location_name,po
st_code,region"}'

POST https://search.api.rioseo.com/search
Authorization: Bearer {{token}}
Content-Type: application/json
{"select":"address_1,address_2,suite,city,fid,lat,lng,location_name,pos
t_code,region"}
```

### API Response:

[ This is equivalent to the default search, *GET https://search.api.rioseo.com/search*.  
You may add and remove fields from this list to change the returned contents.  
You may use any values returned by the *GET https://search.api.rioseo.com/fields/select* request. ]

This API call performs a filtered search, demonstrating every available parameter.

### API Request:

```
curl -X POST "https://search.api.rioseo.com/search/2"
-H "Authorization: Bearer {{token}}" -H "Content-Type:
application/json"
-d
'{"select":"...", "from":"...", "where":"...", "orderBy":"fid", "sortDesc":false,
"page":2, "limit":50}'

POST https://search.api.rioseo.com/search
Authorization: Bearer {{token}}
Content-Type: application/json
{

"select":"address_1,address_2,suite,city,fid,lat,lng,location_name,post
_code,region",
  "from":"rioseo.com",
  "where": {
    "city_lc": "san diego",
    "boolean.open_saturdays": 1,
    "boolean.live_data.custom": 0
  },
  "orderBy": "fid",
  "sortDesc": false,
```

```
    "page":2,  
    "limit":50  
  }
```

**API Response:**

[ This search would return address information for the 51<sup>st</sup> – 100<sup>th</sup> store locations in San Diego that are Open Saturdays, and don't have a *custom* live data message. For multi-domain clients, the *from* field can be used to indicate which domain you're querying. It accepts both numeric ids, and the domain name. ]

At this point, you have seen the full capability of the API. You can mix and match any of the POST parameters you need, in any order. 1 and 0 are used to indicate true and false. For boolean fields, the *true* | *false* value is determined by JavaScript boolean value rules, so empty strings are false, but non-empty strings are true. 0 is false, non-zero numbers are true.