



Right On Interactive API Product Documentation

version 0.9 (beta)

[Overview](#)

[Authenticating](#)

[Signing](#)

[Timestamp and Nonce](#)

[Making API Calls](#)

[Rate Limits](#)

[Caching](#)

[Specifying Search Criteria](#)

[String Data Types](#)

[Simple Search](#)

[Wildcard Search](#)

[Is Null or Empty and Is Not Null or Empty](#)

[Numeric Data Types](#)

[Simple Search](#)

[Range Search](#)

[Date Data](#)

[Simple Search](#)

[Range Search](#)

[DateTime Data](#)

[Simple Search](#)

[Range Search](#)

[API Endpoint List](#)

[Generate API Key](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[Searching the API Logs](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[Examples](#)

[Get Table Schema](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Get Table Column Schema](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Get Single Table Row](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Insert Table Rows](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Request Format](#)

[JSON](#)

[XML](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Update Single Table Row](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Request Format](#)

[JSON](#)

[XML](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Update Multiple Table Rows](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Request Format](#)

[JSON](#)

[XML](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Searching Table Rows](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Delete Single Table Row](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Delete Multiple Table Rows](#)

[Details](#)

[Description](#)

[Query Parameters](#)

[Request Format](#)

[JSON](#)

[XML](#)

[Response Format](#)

[JSON](#)

[XML](#)

[Object Definitions](#)

[Table](#)

[Column](#)

[Search Results](#)

[Log Entry](#)

Overview

Authenticating

In order to successfully make API calls, you will need an API key. API keys are created and distributed by Right On Interactive. If you do not have an API key, but would like one, please contact your ROI Client Success Manager or email support@rightoninteractive.com to obtain one.

Once you have an API key, each API call you make will need to include the key. Here is an example of retrieving a row from your Contacts table with a sample API key included

```
GET /api/v1/Contacts/1001?apiKey=mTil9lLOjAVzWJSj
```

If you do not include an API key in your call, you will receive a 403 response with the following message:

```
{
  "Message": "Invalid or missing API key.",
  "Details": "Please refer to ... for help with this error."
}
```

If the API key you provided is incorrect, you will receive a 403 response with the following message:

```
{
  "Message": "The API key provided could not be decrypted."
  "Details": "Please refer to ... for help with this error."
}
```

In the event of an API key related error, please contact support@rightoninteractive.com for assistance.

Signing

In addition to the API key, you will also receive an API secret key. This secret key should be kept secure and should not be sent with any API calls. Your secret key will be used to sign your API calls.

In order to ensure that you and only you are making API calls on your behalf, we require that you sign each API call using a cryptographically secure algorithm. This hashed signature should be placed in the request header along with a timestamp and nonce. In order to create the hashed signature, append the following items separated by a carriage return and newline:

The HTTP method in all capital letters (GET, POST, PUT, or DELETE).
The API endpoint in all lower-case letters.
The query string starting with the ? symbol.
The Timestamp and Nonce header values in lower-case letters.
The request body (POST, PUT, or DELETE).
Your secret key.

Once these items are appended, they should be hashed using SHA2 and the result base64 encoded.
Here two examples:

```
GET
/api/v1/tables/schema
?apiKey=myApiKey
nonce=acc2715b-a2a5-4f22-8806-62c50a5a33ae&timestamp=2014-02-13t15:35:00.6045320z

mySecretKey
```

Which would result in the hash: nyEe+9tvPVPEH4t+X95cqAvVbn3eEBJF1+kIgeJBpg4=

```
POST
/api/v1/tables/schema
?apiKey=myApiKey
nonce=3b250d4b-1d77-4402-95d6-d91d97f22d6f&timestamp=2014-02-13t15:36:03.8521496z
Sed semper, diam sed consequat pretium, sem eros sodales urna, vel auctor.

mySecretKey
```

which would result in the hash: hDSXtxSB1t2/J71ASilyBmAshGk1n7ko8Fdsg6nP1jw=

Timestamp and Nonce

In order to ensure that your API calls are not being captured, altered, and resent without your knowledge, we have implemented two additional security measure.

A Timestamp request header must be included in each API. The value of this header indicates when the request was created and must be an ISO 8601 formatted UTC time zone date/time string. We will reject requests which were created too far in the past, so it is important to keep the clock of the computer generating requests in sync with NTP.

Additionally, a Nonce request header must also be attached to each API call. We will use this value to determine whether a request has been submitted multiple times. Therefor each Nonce value must be unique. Any approach which produces a relatively random alphanumeric string should be OK here.

Making API Calls

Rate Limits

API calls are limited to 10 calls per 60 seconds. All endpoints are included in the limit. If you exceed the rate limit, the response will be a 403 Forbidden and the body of the response will be in the following format:

```
{
  "Message": "Rate limit exceeded.",
  "Limit": 10,
  "Reset": "60 seconds",
  "Details": "Please refer to ... for help with this error."
}
```

Caching

Most GET calls are cached for 24 hours. The exception to this is the search API paging URLs which are not cached at all. This means if you add a column to a table through our website after requesting the column information from the API, subsequent calls to the API may not reflect the latest changes to that table.

We plan to improve our caching mechanism soon so that this inconsistency is fixed.

Specifying Search Criteria

Search criteria can appear in two places: query parameters for GET calls, and in the body of the request in all other request types. For this reason, we show the JSON version of the criteria next to the query string version in the examples below.

String Data Types

For the most part, when searching string data types, a simple case-insensitive compare is used. You can include a wildcard symbol (*) anywhere in the search term to perform a case-insensitive wildcard search. The following are some examples of string search terms:

Simple Search

Name: "John Doe"

or

Name=John Doe

Compare the name column in the specified table and find all rows whose Name column is equal to "John Doe". This search criterion will match the following strings:

John Doe
john doe
jOhN DoE

Wildcard Search

Name: `"*nat*"` or Name=`*nat*`

Compare the name column in the specified table and find all rows whose Name column contains the consecutive characters N, A, and T. This search criterion will match the following strings:

Nathan Clark
Natalie Hart
Jake Natwick

Is Null or Empty and Is Not Null or Empty

In order to search a column for null or empty values, you should format your query like this:

Address: `""` or Address=`=`

This will return results that are null in the database (there is no value) or where the value is blank (empty string).

In order to search a column for non-null and non-empty values only, you should format your query like this:

Address: `"*"` or Address=`*`

This uses the same wildcard symbol as before. Only rows whose column value is not null or empty will be returned.

Numeric Data Types

When searching numeric data types, you can do a simple comparison or use a range operator.

Compare the BirthDate column with the given date and return all rows that match it exactly.

Range Search

```
JoinDate: "1/1/2014_1/1/2015"
```

or

```
JoinDate=1/1/2014_1/1/2015
```

Compare the JoinDate column in the specified table and find all rows whose JoinDate value is greater than or equal to (\geq) 1/1/2014 and less than ($<$) 1/1/2015.

DateTime Data

Similar to searching floating point numbers, searching for an exact date/time value is problematic and can cause confusion. To this end, we have attempted to simplify date/time searching.

Simple Search

Because of the problems involved with comparing two date/time data fields for equality, a simple search is converted into a range operator for you. Here is an example:

```
CreatedDate: "2010-05-14T22:34:32.763"
```

or

```
CreatedDate=2010-05-14T22:34:32.763
```

Behind the scenes, this is converted into the following range operator search:

```
CreatedDate: "2010-05-14_2010-05-15"
```

Compare the CreatedDate column in the specified table and find all rows whose CreatedDate value is greater than or equal to (\geq) 5/14/2010 and less than ($<$) 5/15/2010.

Notice that the time component was stripped out of the original value when converting to a range search. This essentially gets all the records created on a single day. If you need to do a date search that includes a time component, please use the range search discussed below.

Range Search

The range search for DateTime data behaves identically to the range search for numeric data types. Here are three examples:

```
CreatedDate: "2010-05-10T12:00:00.00_2010-05-11T12:00:00.00"
```

or

```
CreatedDate=2010-05-10T12:00:00.00_2010-05-11T12:00:00.00
```

Compare the CreatedDate column in the specified table and find all rows whose CreatedDate value is greater than or equal to (\geq) noon on 5/10/2010 and less than ($<$) noon on 5/11/2010.

```
CreatedDate: "2013-01-01_" or CreatedDate=2013-01-01_
```

Compare the CreatedDate column in the specified table and find all rows created since midnight January 1st, 2013.

```
CreatedDate: "_2013-07-05T06:00:00.00"
```

or

```
CreatedDate=_2013-07-05T06:00:00.00
```

Compare the CreatedDate column in the specified table and find all rows created before 6:00 AM on July 5th, 2013.

API Endpoint List

Generate API Key

Details

Verb	GET
URL(s)	/api/v1/Admin/GenerateKey /api/v1/Admin/GenerateKey.json /api/v1/Admin/GenerateKey.xml

Description

This API can be used to generate API keys for a user and client. This endpoint is only accessible from the local server (IP address 127.0.0.1 or the server's IP address). The API key returned is required for all other endpoints.

Trying to access this API from an unauthorized location will produce a 403 error with the following message:

```
This endpoint is not accessible from your current location.
```

Query Parameters

Name	Description	Default Value	Required
ClientID	The ID of the client for which the token will be created.	None	Yes
UserID	The ID of the user for whom the token will be created.	None	Yes
Expires	The date on which the generated token will expire.	One year from the creation date	No

Response Format

JSON

```
{
  "PublicKey": "315ce1ddcf55468d958756784200c0f2",
  "SecretKey": "6fbc45f992e4d19ba3929c94f783d64",
  "ClientID": 100,
  "UserID": 100,
  "RateLimit": 20,
  "RateWindow": 1,
  "Expires": "2015-02-25T14:48:30.2949281Z"
}
```

Searching the API Logs

Details

Verb	GET
URL(s)	<code>/api/v1/Admin/Logs</code> <code>/api/v1/Admin/Logs.json</code> <code>/api/v1/Admin/GenerateKey.xml</code>

Description

This API can be used to query the logs generated from previous API calls.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes
Limit	The number of search results to return.	50	No
IPAddress	The IP address from which the API call originated.	None	No
Url	The URL to the API endpoint.	None	No
HttpVerb	GET, PUT, POST, or DELETE	None	No
RequestHeaders	The request header sent to the server.	None	No
RequestQueryString	The query string parameters for the request.	None	No
RequestBody	The request body if the request was a PUT, POST, or DELETE.	None	No
ResponseHeaders	The response headers returned from the server.	None	No

ResponseBody	The response body returned from the server.	None	No
ResponseCode	OK, Bad Request, Forbidden, Not Found, Internal Server Error, etc.	None	No
Exception	The exception message with a description of what caused the error on the server.	None	No
RuntimeMS	The amount of time in milliseconds it took for the server to respond to the request.	None	No
Timestamp	The date and time the API call was made. This will always be a time in the Eastern Time zone.	None	No

Response Format

JSON

```
[
  {
    "ClientID": 182,
    "UserID": 1,
    "IPAddress": ":::1",
    "Url": "/api/v1/Tables/Schema",
    "HttpVerb": "GET",
    "RequestHeaders": "<<Truncated for Brevity>>",
    "RequestQueryString": "?apiKey=<<Redacted>>",
    "RequestBody": "",
    "ResponseHeaders": "ETag: \"2341234\"\\r\\nCache-Control: max-age=100\\r\\n",
    "ResponseBody": "<<Truncated for Brevity>>",
    "ResponseCode": "OK",
    "Exception": "",
    "RuntimeMS": 1590,
    "Timestamp": "2014-01-31T13:24:19.7434836-05:00",
    "ItemName": "2fdadf13-0a1e-4f88-868c-1769f753b3e3"
  },
  {
    "ClientID": 182,
    "UserID": 1,
    "IPAddress": ":::1",
    "Url": "/api/v1/Tables/bad table name/Schema",
    "HttpVerb": "GET",
    "RequestHeaders": "<<Truncated for Brevity>>",
    "RequestQueryString": "?apiKey=<<Redacted>>",
    "RequestBody": null,
    "ResponseHeaders": null,
    "ResponseBody": null,
  }
]
```

```
    "ResponseCode": null,  
    "Exception": "The column information for that table could not be found.",  
    "RuntimeMS": 0,  
    "Timestamp": "2014-02-06T14:27:05.2551025-05:00",  
    "ItemName": "138e239a-8bfe-4999-8d2a-9dee2027a9a9"  
  }  
]
```

Examples

Return all API calls made today (if today is February 6th, 2014):

```
/api/v1/Admin/Logs&Timestamp=2/6/2014_2/7/2014
```

Return all API calls that took longer than a second to return:

```
/api/v1/Admin/Logs?RuntimeMS=1000_
```

Return all API calls in a given week that resulted in an exception:

```
/api/v1/Admin/Logs&Timestamp=2/2/2014_2/10/2014&Exception=*
```


Get Table Schema

Details

Verb	GET
URL(s)	/api/v1/Tables/Schema /api/v1/Tables/Schema.json /api/v1/Tables/Schema.xml

Description

This API will return a list of table objects containing information for all the tables you have access to. Optionally, the column information for each table can also be included.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes
Columns	Flag indicating if the table's column data should be included. Acceptable values are <code>yes</code> and <code>no</code> .	no	No

Response Format

The `Columns` field of the tables will be `null` or blank if the `Columns` query parameter was not `yes`.

JSON

```
[
  {
    "Name": "Company",
    "Description": "Company",
    "Created": "2010-05-14T22:34:32.763",
    "Updated": "2012-04-11T13:59:44.547",
    "ReadOnly": false,
    "Columns": [
      {
        "Name": "ID",
        "Description": null,
        "DataType": "int",
        "Length": 4,
        "DefaultValue": null,
        "Required": true
      }
    ]
  }
]
```

```
    },
    {
      "Name": "Name",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "Address1",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "Address2",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "City",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "State",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "ZipCode",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "Country",
      "Description": null,
      "DataType": "nvarchar",
      "Length": 50,
      "DefaultValue": null,
      "Required": false
    },
    {
      "Name": "CreatedDate",
```

```

        "Description": "Created Date",
        "DataType": "datetime",
        "Length": 8,
        "DefaultValue": "(getdate())",
        "Required": true
    },
    {
        "Name": "LastModifiedDate",
        "Description": "Last Modified Date",
        "DataType": "datetime",
        "Length": 8,
        "DefaultValue": "(getdate())",
        "Required": true
    },
]
},
{
    "Name": "Contacts",
    "Description": "Testing",
    "Created": "2010-05-14T22:34:32.81",
    "Updated": "2013-01-15T13:48:59.57",
    "ReadOnly": false,
    "Columns": [
        {
            "Name": "Id",
            "Description": null,
            "DataType": "int",
            "Length": 4,
            "DefaultValue": null,
            "Required": true
        },
        {
            "Name": "CompanyID",
            "Description": null,
            "DataType": "int",
            "Length": 4,
            "DefaultValue": null,
            "Required": true
        },
        {
            "Name": "Name",
            "Description": null,
            "DataType": "nvarchar",
            "Length": 50,
            "DefaultValue": null,
            "Required": false
        },
        {
            "Name": "Email",
            "Description": null,
            "DataType": "nvarchar",
            "Length": 50,
            "DefaultValue": null,
            "Required": false
        },
        {
            "Name": "CreatedDate",
            "Description": "Created Date",
            "DataType": "datetime",

```

```
        "Length": 8,  
        "DefaultValue": "(getdate())",  
        "Required": true  
    },  
    {  
        "Name": "LastModifiedDate",  
        "Description": "Last Modified Date",  
        "DataType": "datetime",  
        "Length": 8,  
        "DefaultValue": "(getdate())",  
        "Required": true  
    },  
    ]  
}  
]
```

XML

Get Table Column Schema

Details

Verb	GET
URL(s)	<code>/api/v1/Tables/{TableName}/Schema</code> <code>/api/v1/Tables/{TableName}/Schema.json</code> <code>/api/v1/Tables/{TableName}/Schema.xml</code>

Description

This API will return a list of column objects for the specified table. The information returned will include the name of the column, a description, the SQL data type, the maximum length in bytes, the formula to generate the default value, and a flag indicating if the column is required for new rows inserted into the table.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes

Response Format

JSON

```
[
  {
    "Name": "Id",
    "Description": null,
    "DataType": "int",
    "Length": 4,
    "DefaultValue": null,
    "Required": true
  },
  {
    "Name": "CompanyID",
    "Description": null,
    "DataType": "int",
    "Length": 4,
    "DefaultValue": null,
    "Required": true
  },
]
```

```
{
  "Name": "Name",
  "Description": null,
  "DataType": "nvarchar",
  "Length": 50,
  "DefaultValue": null,
  "Required": false
},
{
  "Name": "Email",
  "Description": null,
  "DataType": "nvarchar",
  "Length": 50,
  "DefaultValue": null,
  "Required": false
},
{
  "Name": "CreatedDate",
  "Description": "Created Date",
  "DataType": "datetime",
  "Length": 8,
  "DefaultValue": "(getdate())",
  "Required": true
},
{
  "Name": "LastModifiedDate",
  "Description": "Last Modified Date",
  "DataType": "datetime",
  "Length": 8,
  "DefaultValue": "(getdate())",
  "Required": true
},
}
]
```

XML

Get Single Table Row

Details

Verb	GET
URL(s)	/api/v1/Tables/{TableName}/{ID}

Description

This API allows you to retrieve the data for a single row in a table identified by the given ID. The response will contain all the fields in the row optionally filtered by the column list parameter.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes
Columns	A list of columns from which to return the value of the specified row.	None	No

Response Format

JSON

```
{
  "ID": 2109167,
  "FirstName": "John W.",
  "LastName": "Doe",
  "Email": "jwdoe@gmail.com"
  "CreateDate": "2013-12-16T18:27:02.34",
  "LastModifiedDate": "2013-12-16T19:23:29.56"
}
```

XML

--

Insert Table Rows

Details

Verb	POST
URL(s)	/api/v1/Tables/{TableName}/Insert

Description

This API allows you to insert data into the specified table. You can insert up to 1,000 objects at a time. The response will contain the ID, CreatedDate, LastModifiedDate, and resource URL for the objects that are created.

This API call is atomic. If there is an error at any time, no objects will be inserted into the table. This guarantees that all the objects in the request body are inserted or that none of them are inserted.

The content-type header must be set with the format of the incoming data: application/json or application/xml.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes

Request Format

The request body must contain a list of object data to insert into the specified table.

JSON

```
POST /api/v1/Tables/Contacts/Insert
[
  {
    "FirstName": "John",
    "LastName": "Doe",
    "Email": "jdoe@gmail.com"
  },
  {
    "FirstName": "Jane",
```



```
    "LastName": "Doe",
    "Email": "jdoe@yahoo.com"
  },
  {
    "FirstName": "Bob",
    "LastName": "Jones",
    "Email": "bjones@hotmail.com"
  },
]
```

XML

Response Format

JSON

```
[
  {
    "ID": 2109167,
    "CreatedDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T18:27:02.34",
    "Url": "api/v1/Tables/Contacts/2109167"
  },
  {
    "ID": 2109168,
    "CreatedDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T18:27:02.34",
    "Url": "api/v1/Tables/Contacts/2109168"
  },
  {
    "ID": 2109169,
    "CreatedDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T18:27:02.34",
    "Url": "api/v1/Tables/Contacts/2109169"
  }
]
```

XML

Update Single Table Row

Details

Verb	PUT
URL(s)	/api/v1/Tables/{TableName}/{ID}/Update

Description

This API allows you to update a single row in the specified table. The table should be identified by its ID field. The fields you want updated should be included in the request body.

The content-type header must be set with the format of the incoming data: application/json or application/xml.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes

Request Format

The request body should contain an object with fields and values. The field names should match the column names in the target table and the values should be convertible to the column data type. Only the fields included in the request body will be modified.

JSON

```
PUT /api/v1/Tables/Contacts/2109167/Update
{
  "FirstName": "John W.",
  "LastName": "Doe",
  "Email": "jwdoe@gmail.com"
}
```

XML

Response Format

JSON

```
{  
  "ID": 2109167,  
  "CreateDate": "2013-12-16T18:27:02.34",  
  "LastModifiedDate": "2013-12-16T19:05:57.01",  
  "Url": "api/v1/Tables/Contacts/2109167"  
}
```

XML

Update Multiple Table Rows

Details

Verb	PUT
URL(s)	/api/v1/Tables/{TableName}/Update

Description

This API allows you to update multiple rows in the specified table. The fields you want updated should be included in the request body. Additionally, a list of criteria should be included in order to determine which rows should be updated.

The content-type header must be set with the format of the incoming data: application/json or application/xml.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes

Request Format

The request body should contain an object which itself contains an object named Fields and an object named Criteria.

The Fields object should contain a list of fields and values. The field names should match the column names in the target table and the values should be convertible to the column data type. Only the fields included in the request body will be modified.

The Criteria object should also contain a list of fields and value. The field names should also match the column names in the target table, but the values will be evaluated in order to segment the table rows into those that will be updated.

JSON

PUT /api/v1/Tables/Contacts/Update

```
{
  Fields:
  {
    "FirstName": "John W.",
    "LastName": "Doe",
    "Email": "jwdoe@gmail.com"
  },
  Criteria:
  {
    "Name": "*doe*",
    "CreatedDate": "1/1/2013_"
  }
}
```

XML

Response Format

JSON

```
[
  {
    "ID": 2109167,
    "CreatedDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T19:23:29.56",
    "Url": "api/v1/Tables/Contacts/2109167"
  },
  {
    "ID": 2109168,
    "CreatedDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T19:23:29.56",
    "Url": "api/v1/Tables/Contacts/2109168"
  },
]
```

XML

Searching Table Rows

Details

Verb	GET
URL(s)	/api/v1/Tables/{TableName}

Description

This API allows you to query the data stored in a table and return rows matching that criteria.

The response will contain URLs you can use to page through the same result set. For the best performance, please use these URLs in lieu of other solutions if you plan to page through large result sets.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes
Limit	The maximum number of rows to return.	50	No
Columns	A list of columns to return from the results. By default all columns are returned.	None	No
Search Terms	A list of name value pairs used to filter the table rows. Please refer to the "Specifying Search Criteria" section for details.	None	No

Response Format

JSON

```
{
  "Total": 5239,
  "Results":
  [
    {
      "ID": 2109167,
      "CreatedDate": "2013-12-16T18:27:02.34",
      "LastModifiedDate": "2013-12-16T19:23:29.56",
      "Url": "api/v1/Tables/Contacts/2109167"
    },
  ],
}
```

```
{
  "ID": 2109168,
  "CreateDate": "2013-12-16T18:27:02.34",
  "LastModifiedDate": "2013-12-16T19:23:29.56",
  "Url": "api/v1/Tables/Contacts/2109168"
},
],
"NextPage": "/api/v1/Tables/Contacts/53c163b2140741348a96f16a4ba0bed5/Next",
"PrevPage": "/api/v1/Tables/Contacts/53c163b2140741348a96f16a4ba0bed5/Previous",
"Expires": "2014-01-13T19:01:43.5601473Z"
}
```

XML

Delete Single Table Row

Details

Verb	DELETE
URL(s)	/api/v1/Tables/{TableName}/{ID}/Delete

Description

This API allows you to delete a single row in the specified table with the given ID. The response will contain an object containing all the fields and values for the deleted table row.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes
Columns	A list of columns from which to return the value of the deleted row.	None	No

Response Format

JSON

```
{
  "ID": 2109167,
  "FirstName": "John W.",
  "LastName": "Doe",
  "Email": "jdoe@gmail.com",
  "CreateDate": "2013-12-16T18:27:02.34",
  "LastModifiedDate": "2013-12-16T19:05:57.01"
}
```

XML

--

Delete Multiple Table Rows

Details

Verb	DELETE
URL(s)	/api/v1/Tables/{TableName}/Delete

Description

This API allows you to delete multiple rows in the specified table. A list of criteria should be included in the request body in order to determine which rows should be deleted. The response will contain a collection of objects containing all the fields and values for the deleted table rows.

The content-type header must be set with the format of the incoming data: application/json or application/xml.

Query Parameters

Name	Description	Default Value	Required
apiKey	Your API key.	None	Yes
Columns	A list of columns from which to return the value of the deleted row.	None	No

Request Format

The request body should contain an object that contains a list of fields and values. The field names should also match the column names in the target table and the values will be evaluated in order to segment the table rows into those that will be deleted.

JSON

```
DELETE /api/v1/Tables/Contacts/Delete
{
  "Name": "*doe*",
  "CreateDate": "1/1/2013_"
}
```

XML

Response Format

JSON

```
[
  {
    "ID": 2109167,
    "FirstName": "John W.",
    "LastName": "Doe",
    "Email": "jdoe@gmail.com",
    "CreateDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T19:23:29.56",
    "Url": "api/v1/Tables/Contacts/2109167"
  },
  {
    "ID": 2109168,
    "FirstName": "Jane",
    "LastName": "Doe",
    "Email": "jdoe@yahoo.com",
    "CreateDate": "2013-12-16T18:27:02.34",
    "LastModifiedDate": "2013-12-16T19:23:29.56",
    "Url": "api/v1/Tables/Contacts/2109168"
  },
]
```

XML

Object Definitions

Table

Name	Description	Data Type
Name	The name of the table.	string
Description	The description of the table.	string
Created	The date/time the table was created.	date/time
Updated	The date/time the table was last updated.	date/time
ReadOnly	Whether this table will accept insert, update, and delete commands.	boolean
Columns	An collection of Column objects representing the columns in this table.	Column[]

Column

Name	Description	Data Type
Name	The name of the table.	string
Description	The description of the table.	string
DataType	The name of the data type that is stored in this columns. Possible values include: nvarchar, varchar, int, float, date, datetime, and bit.	string
Length	If the datatype of the column is varchar or nvarchar, this is the maximum number of characters that can be stored in the column.	integer
DefaultValue	This is the value that will be used for this column during an insert if there is no value supplied.	string
Required	If this field is true, a value must be supplied for this column during and insert.	boolean

Search Results

Name	Description	Data Type
Total	The total number of results that match the criteria give. The Results field will be a subset of these results.	integer
Results	A collection of objects representing the current page of results.	object[]
NextPage	A URL that can be used to retrieve the next page of results using the same search criteria.	string
PrevPage	A URL that can be used to retrieve the previous page of results using the same search criteria.	string
Expires	The date/time on which the NextPage and PrevPage URLs will expire.	date/time

Log Entry

Name	Description	Data Type
ClientID	The ID assigned to your client internally.	integer
UserID	The ID of the user that executed the API call.	integer
IPAddress	The IP address from which the API call originated.	string
Url	The URL to the API endpoint.	string
HttpVerb	GET, PUT, POST, or DELETE	string
RequestHeaders	The request header sent to the server.	string
RequestQueryString	The query string parameters for the request.	string
RequestBody	The request body if the request was a PUT, POST, or DELETE.	string
ResponseHeaders	The response headers returned from the server.	string
ResponseBody	The response body returned from the server.	string

ResponseCode	OK, Bad Request, Forbidden, Not Found, Internal Server Error, etc.	string
Exception	The exception message with a description of what caused the error on the server.	string
RuntimeMS	The amount of time in milliseconds it took for the server to respond to the request.	integer
Timestamp	The date and time the API call was made. This will always be a time in the Eastern Time zone.	date/time
ItemName	The unique name given to the log entry internally.	string