Metrics API Manual

Blake Bockrath

Lee Keeler

Cole Pack

Jonathan Schallon

Tyler Simpson

Habtamu Tegegne

Zack Vickers

Contents

[INTRODUCTION: 2](#_Toc53423425)

[FORMATTING: 2](#_Toc53423426)

[Running API 2](#_Toc53423427)

[CREATE: 3](#_Toc53423428)

[READ: 4](#_Toc53423429)

[UPDATE: 5](#_Toc53423430)

[DELETE: 6](#_Toc53423431)

[APPENDIX 7](#_Toc53423432)

# INTRODUCTION:

The Metrics API is an interface to access the Metric database over the http web connection protocol. This API provides basic CRUD(Create Read Update Delete) functionality. The API is written using the .NET API format.

# FORMATTING:

Primary Key Format:

* Single Numeric: …/key
* Single String: …/’key’
* Composite Numeric: …/key&key
* Composite Mixed: …/key&’key’
* See Table 1 in Appendix for key order within composite keys.

Data format in HTTP body:

{

    {FieldName}: {Data},

…

{FieldName}: {Data}

}

* Data requires surrounding double quotes if non-numeric.
* FieldNames always use quotes.
  + Ex. “FieldName”: “DateorText”,
  + Ex. “FieldName”: Numeric,

# Running API

Local Machine

* Via Visual Studio
  + 

Domains[[1]](#footnote-1)

* Local:localhost:5002
* Remote:https://segfault.asuscomm.com:9300

# CREATE:

HTTP POST [https://segfault.asuscomm.com:9300/api/Metrics/{table}](https://segfault.asuscomm.com:9300/api/Metrics/%7btable%7d)

Body:

{

    {FieldName}: {Data},

…

{FieldName}: {Data}

}

Body should contain primary key fields at minimum. Any other fields may be entered as well.

Field order does not matter.

Auto Commit on success.

Example URL and Body:

https://segfault.asuscomm.com:9300/api/Metrics/Customer

Body:

{

        "CUSTOMERID": 2,

        "EMAIL": "waldo@gmail.com",

        "FIRSTNAME": "Waldo",

        "LASTNAME": "Wheris",

        "ADDRESS": "101 somewhere, lost",

        "ZIP": "37601",

        "STATE": "TN",

        "PAYMENTMETHOD": "IsBroke"

    }

table = paint

Primary Key = “CUSTOMERID”: 2

Result: A new record is created in the Customer table with the fields listed in the body. Success on HTTP OK received.

# READ:

HTTP GET https://segfault.asuscomm.com:9300/api/Metrics/{table}/{primary key}

Body: NULL

Entering the desired table without the primary key will return all items in the table.

Entering table and primary key will return the item in the table with that primary key.

Example URL:

https://segfault.asuscomm.com:9300/api/Metrics/Customer/1

table = Customer

Primary key:

CustomerID = 1

How to access in .Net C#

<https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client>

# UPDATE:

HTTP PUT https://segfault.asuscomm.com:9300/api/Metrics/{table}/{primary key}

Body:

{

    {FieldName}: {Data},

{FieldName}: {Data}

}

The body only requires the fields to be changed, not all the fields in the table.

Update only affects a single item. I.E. Incomplete Composite keys will fail.

Changes are automatically committed.

Example URL and Body:

https://segfault.asuscomm.com:9300/api/Metrics/customer/1

Body:

{

    "PAYMENTMETHOD": “IsNotBroke”

}

table = customer

Primary key:

customerID = 1

Result: The data in the field “PaymentMethod” for the item matching the primary key is changed to ”IsNotBroke”(previous value was “IsBroke”). Success on HTTP OK received.

# DELETE:

HTTP DELETE https://segfault.asuscomm.com:9300/api/Metrics/{table}/{primary key}

Body: NULL

Must enter complete primary key.

Partial composite keys may result in deleting all matching records.

Example URL:

https://segfault.asuscomm.com:9300/api/BikeShop/customer/1

table = customer

Primary key :

customerID = 1

Result: The record matching the composite key is deleted.

# APPENDIX

**Table 1 – No Tables exist with Composite Keys**

1. Example URLs listed in the manual are for API running on a remote machine. Use localhost:5000 for local API. [↑](#footnote-ref-1)