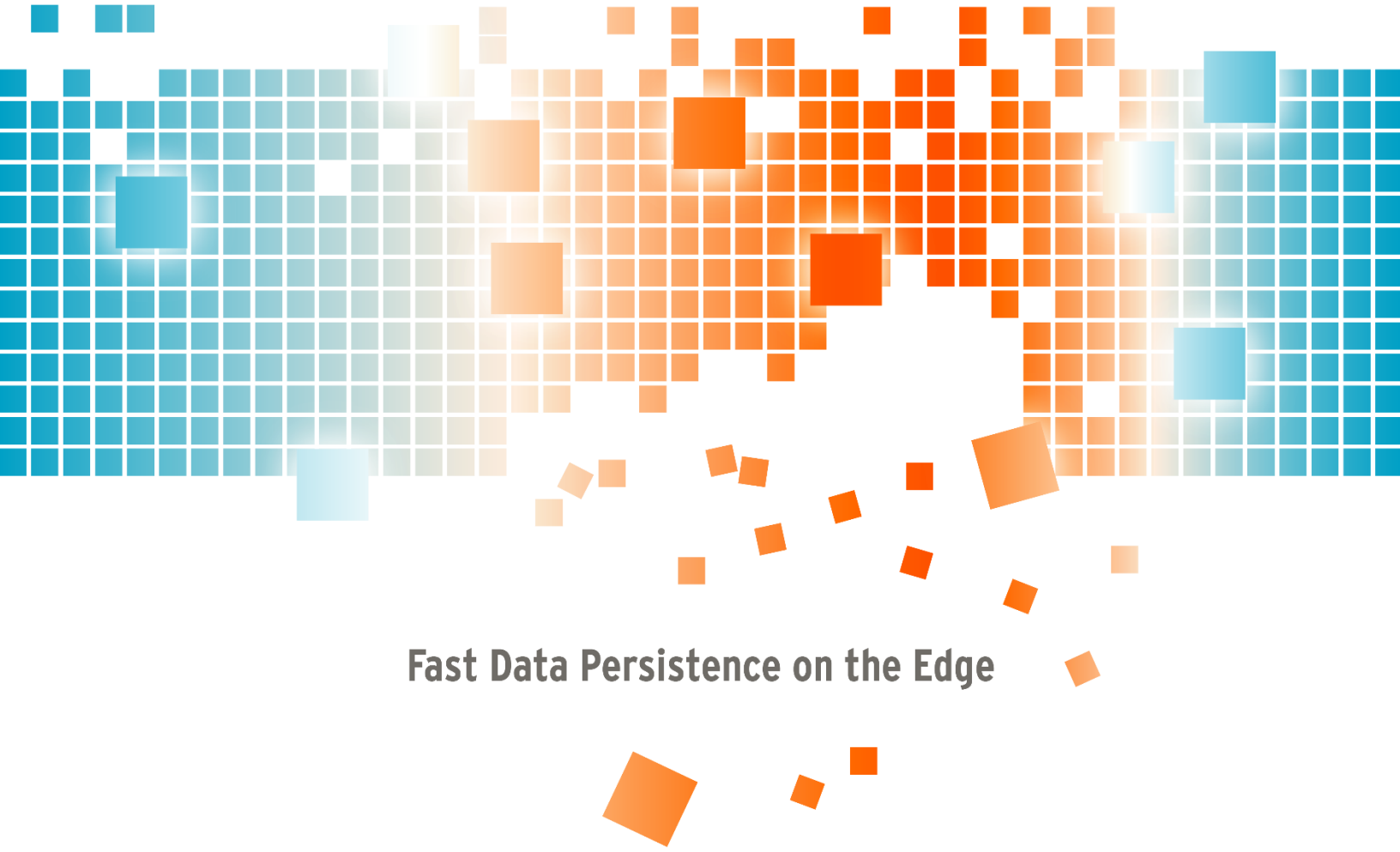
c-treeEDGE
IoT DATABASE



Fast Data Persistence on the Edge

Developer's Guide

c-treeEDGE REST API Tutorial

Contents

1. c-treeEDGE REST API Tutorial

Introduction

c-treeEDGE introduces a new REST API to create, read, update and delete tables, indexes and records.

This tutorial will show you how to use it with a free and open-source command-line tool called **curl**.

1.1 Preparation

1. Make sure your c-treeEDGE MicroServer is running.
   You can test this using the FairCom SQL or ISAM Explorer.
2. Make sure you have a database called `test`.
   Make sure your c-treeEDGE REST server is running on port 8081.
   You can test this by using this URL in your browser:
   http://localhost:8081/ctree/api/v1/openapi

   Note: If you have changed the port (using `listening_http_port` or `listening_https_port`, as described in *HTTP Port* (page 2, *https://docs.faircom.com/doc/c-treeEDGE_DevelopmentGuide/75845.htm*)), you will need to change the URLs in this tutorial accordingly.
3. Make sure **curl** is installed and working.
   You can test his typing in the command line:
   `curl --version`
4. Create a test folder where we can create some JSON files.

All versions of OS X starting with Jaguar (released 2002) come with **curl/libcurl** installed.
curl ships with Windows 10 build 1803 (released early May 2018).
For older versions of Windows, the official download location is:
https://curl.haxx.se/dlwiz/?type=bin&os=Win32&flav=-
curl is usually installed by default on all Linux distributions.

This tutorial assumes that you have the REST API server running on the same machine as you run **curl**.
If this is not the case, you will have to modify the **curl** calls and change localhost to the host name or IP where the REST API server runs.

This tutorial assumes that you have the REST API server running without SSL/HTTPS.
If this is not the case you will have to modify the **curl** calls and change *http://* to *https://*
You may have to adapt the port number, too.
This may also depend on valid certificate files used by the server.

HTTP Port

c-treeEDGE includes a web server that allows HTTP access for the REST API and browser-based tools.

The server is pre-configured to listen for HTTP connections on port 8081. If you have a port conflict, you can change it to an unused port.

A different port is used to enable HTTPS connections (see below).

To change the port used by your server, include the following keyword in the `SUBSYSTEM HTTP SERVER` portion of your *ctsrvr.cfg* file:

```
SUBSYSTEM HTTP SERVER
{
    ENABLED YES
    listening_http_port 8081
    listening_https_port 8443
    ssl_certificate fccert.pem
}
```

`listening_http_port 8081` – Sets the port number used by the HTTP server for the REST API and web tools. `8081` is the default; you can change it to an available port.

`listening_https_port 8443` – Sets the port used with SSL communication. You will need to use the ssl_certificate keyword to enable HTTPS.

`ssl_certificate fccert.pem` – Enables SSL for a HTTPS connection. *fccert.pem* is the name of the certificate file.

1.2 Validating Your Connection

We'd like to make sure your connection is working by checking to see if you have any existing tables. The expectation is you won't have, so after executing the following command, we would expect you to either see an empty result set, or if you do have tables, then you see your existing tables:

```
curl -u admin:ADMIN http://localhost:8081/ctree/api/v1/table/test
```

This should result in something similar to:

```
{"_tables":[]}
```

or

```
{"_tables":["test1","test2","test3"]}
```

1.3 Creating a Table

Change into the test folder and create a new file called *create_table.txt*.

Insert this text into the new file:

```
{"fields": [
    {
        "name": "id",
        "type": "INTEGER",
        "length": 4
    },
    {
        "name": "name",
        "type": "VARCHAR",
        "length": 128
    },
    {
        "name": "author",
        "type": "VARCHAR",
        "length": 128
    },
    {
        "name": "country",
        "type": "VARCHAR",
        "length": 48
    },
    {
        "name": "language",
        "type": "VARCHAR",
        "length": 548
    },
    {
        "name": "published",
        "type": "INTEGER",
        "length": 4
    },
    {
        "name": "pages",
        "type": "INTEGER",
        "length": 4
    }
]}
```

Now execute the following command (in a single line):

```
curl -u admin:ADMIN -d @create_table.txt http://localhost:8081/ctree/api/v1/table/test/books
```

Check your database and you should see a new table with the columns specified in the JSON file.

1.4 Creating an Index

Inside the test folder create a new file called *create_index.txt*.

Insert this text into the new file:

```
{"fields":[{"name": "country", "ascending": true}],"unique":false}
```

Now execute the following command (in a single line):

```
curl -u admin:ADMIN -d @create_index.txt
http://localhost:8081/ctree/api/v1/index/test/books/c_index
```

Check your table and you should see a new index using the column specified in the JSON file.

1.5 Inserting Records

Inside the test folder create a new file called *create_records.txt*.

Insert this text into the new file:

```
[{
  "name": "Anna Karenina",
  "author": "Leo Tolstoy",
  "country": "Russia",
  "language": "Russian",
  "published": 1877,
  "pages": 864
},
{
  "name": "War and Peace",
  "author": "Leo Tolstoy",
  "country": "Russia",
  "language": "Russian, with some French",
  "published": 1869,
  "pages": 1225
},
{
  "name": "Adventures of Huckleberry Finn",
  "author": "Mark Twain",
  "country": "USA",
  "language": "English",
  "published": 1884,
  "pages": 366
},
{
  "name": "To Kill a Mockingbird",
  "author": "Harper Lee",
  "country": "USA",
  "language": "English",
  "published": 1960,
  "pages": 281
},
{
  "name": "Nineteen Eighty-Four",
  "author": "George Orwell",
  "country": "United Kingdom",
  "language": "English",
  "published": 1949,
  "pages": 368
},
{
  "name": "The Great Gatsby",
  "author": "F. Scott Fitzgerald",
  "country": "USA",
  "language": "English",
```

```
  "published": 1925,
  "pages": 180
},
{
  "name": "Moby Dick",
  "author": "Herman Melville",
  "country": "USA",
  "language": "English",
  "published": 1851,
  "pages": 822
}]
```

Now execute the following command (in a single line):

```
curl -u admin:ADMIN -d @create_records.txt http://localhost:8081/ctree/api/v1/record/test/books
```

This should result in:

```
{"_ids":[1,2,3,4,5,6,7]}
```

Check your table and you should see 7 records matching the contents of the JSON file.

1.6 Reading a Single Record

To read a single record execute the following command:

```
curl -u admin:ADMIN http://localhost:8081/ctree/api/v1/record/test/books/1
```

This should result in:

```
{"id":null,"name":"Anna Karenina","author":"Leo
Tolstoy","country":"Russia","language":"Russian","published":1877,"pages":864}
```

1.7 Querying Records

Inside the test folder create a new file called *query_records.txt*.

Insert this text into the new file:

```
{
    "find": {
        "country": {
            "operator": "=",
            "value": "USA"
        }
    }
}
```

Now execute the following command (in a single line):

```
curl -u admin:ADMIN -d @query_records.txt
http://localhost:8081/ctree/api/v1/query/test/books/c_index?top=100
```

This should result in:

```
{
```

```
    "_records": [{
            "_id": 3,
            "id": null,
            "name": "Adventures of Huckleberry Finn",
            "author": "Mark Twain",
            "country": "USA",
            "language": "English",
            "published": 1884,
            "pages": 366
    }, {
            "_id": 4,
            "id": null,
            "name": "To Kill a Mockingbird",
            "author": "Harper Lee",
            "country": "USA",
            "language": "English",
            "published": 1960,
            "pages": 281
    }, {
            "_id": 6,
            "id": null,
            "name": "The Great Gatsby",
            "author": "F. Scott Fitzgerald",
            "country": "USA",
            "language": "English",
            "published": 1925,
            "pages": 180
    }, {
            "_id": 7,
            "id": null,
            "name": "Moby Dick",
            "author": "Herman Melville",
            "country": "USA",
            "language": "English",
            "published": 1851,
            "pages": 822
    }]
}
```


1.8 Deleting a Table

If you would like to clean up here is the call to delete the demo table.

> **Please Note**: Be very careful to use the exact database and table name. **This command is very dangerous**. If you use it on a production database please make sure you have a recent and up-to-date backup just to be safe.

Execute the following command (in a single line):

```
curl -u admin:ADMIN -X DELETE http://localhost:8081/ctree/api/v1/table/test/books
```

1.9 OpenAPI

This resource represents the OpenAPI specification for this REST API.

Type: GET

URL: `/ctree/api/v1/openapi`

For a complete overview of the API please take a look at:

`http://localhost:8443/ctree/api/v1/openapi`

The above URL can be used in this OpenAPI viewer:

`https://petstore.swagger.io/`

To learn more about OpenAPI please follow this link:

`https://github.com/OAI/OpenAPI-Specification`

1.10 REST default date and time string format

The c-treeDB default date and time format is set to:

- `CTDATE_MDCY`
- `CTTIME_HMST`

This results in the following string format:

`MM/DD/CCYY h|hh:mm:ss.ttt` (24 h) (`ttt` are milliseconds), hours can be either 1 or two digits.

The REST API expects and returns dates, times, and timestamps in the above string format.

Copyright Notice

Trademarks

11/16/2018

2. Index