Lab: HTTP API

version: 1.01

An introduction to some of the available operations that can be performed using HTTP requests.

OVERVIEW

EventStoreDB provides http endpoints for a number of user and admin operations.

You might use a tool like Postman or http code libraries to access this functionality in production.

Command line tools such as cURL are also commonly used for prototyping applications that use http endpoints.

This exercise is comprised of the following components:

1. Structure of this demo
2. HTTP tools
3. Review of http and format of http request in httpyac
4. Install the httpyac plugin for VScode
5. Run some examples
6. httpyac format compared to cURL

1. Structure of this demo

This Demo uses the httpyac plugin for VScode to demonstrate available http endpoints in EventStoreDB.

The assumption is an unsecured cluster running locally.

If you need to start your cluster follow these steps:

Switch to the directory `C:\esdb`

```
cd C:\esdb
```

And executing this command.

The instructions assume an insecure cluster, so make sure the “–insecure” setting is used.

```
.\EventStore.ClusterNode.exe "--dev" "--insecure"
```

2. HTTP Tools Overview and intro to httpyac

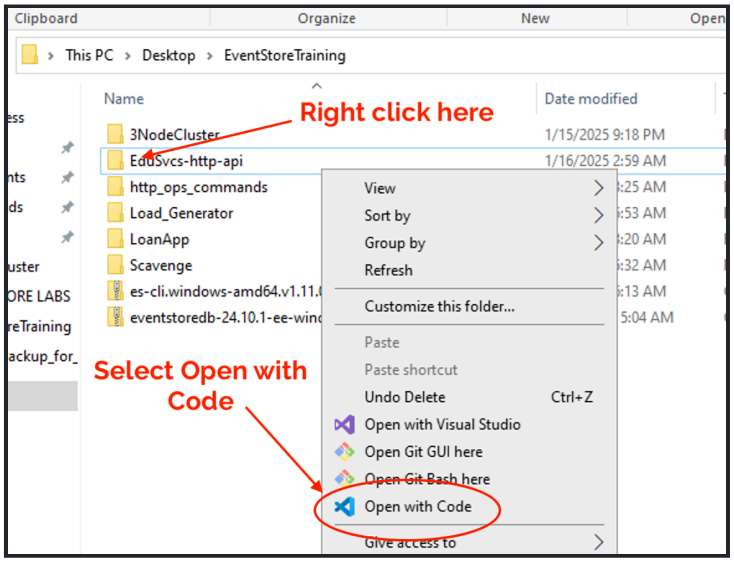
Curl is often used for testing http requests, and Postman is a tool often used for testing and production. For this training content, the following issues were considered

1. Curl can be cumbersome, and command execution can be prone to typos and other issues.
2. Postman requires an account to use, and asking students to create a Postman account is not reasonable.

The solution for this training is to use the VScode plugin httpyac, https://marketplace.visualstudio.com/items?itemName=anweber.vscode-httpyac

Httpyac is designed as a test management tool, but it has a simple text-based representation of the http request, and no login or registration is required.

3. Open the HTTP examples folder in vscode



4. A quick review of HTPP and the format of a httpyac command

HTTP Request details

An HTTP request has:

- a request line
- optional header fields
- optional message body.

Request-Line

A request line consists of a request method, target, and the HTTP protocol version. If the request method is omitted, 'GET' will be used as a default. The HTTP protocol version can also be omitted.

See line 1, of `00_basic_get.http` in the `EventstoreDB_examples/examples_insecure_cluster` folder.

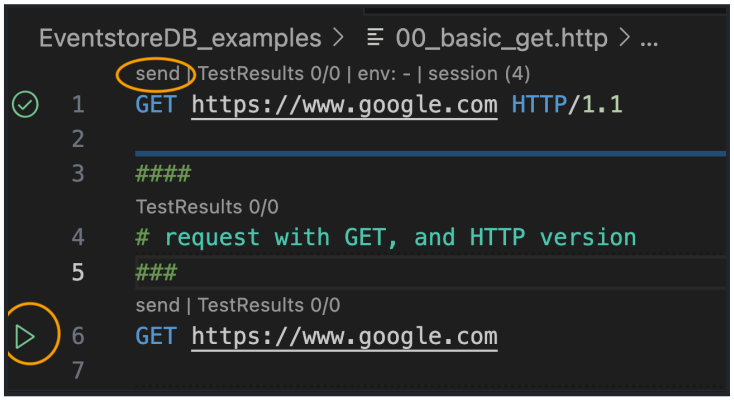
This is a line with the request type GET included, to the host `www.google.com,` using HTTP version 1.1

```
GET https://www.google.com HTTP/1.1
```

Note that if the request type was not included GET would be assumed.

If the protocol is ommitted it would default to HTTP/1.1

The file contains examples of various formats for what resolves to the same request. In vscode, with the httpyac plugin installed, you can run each line by clickin on `send` above the line, or clicking on the green arrow next to the line.



Query-String

Query string can be part of the request line.

```
GET https://google.com/?q=EventStore
```

Or the query string can be on the next line.

```
GET https://google.com/
    ?q=EventStore
```

Headers

Headers are entered one per line following the request.

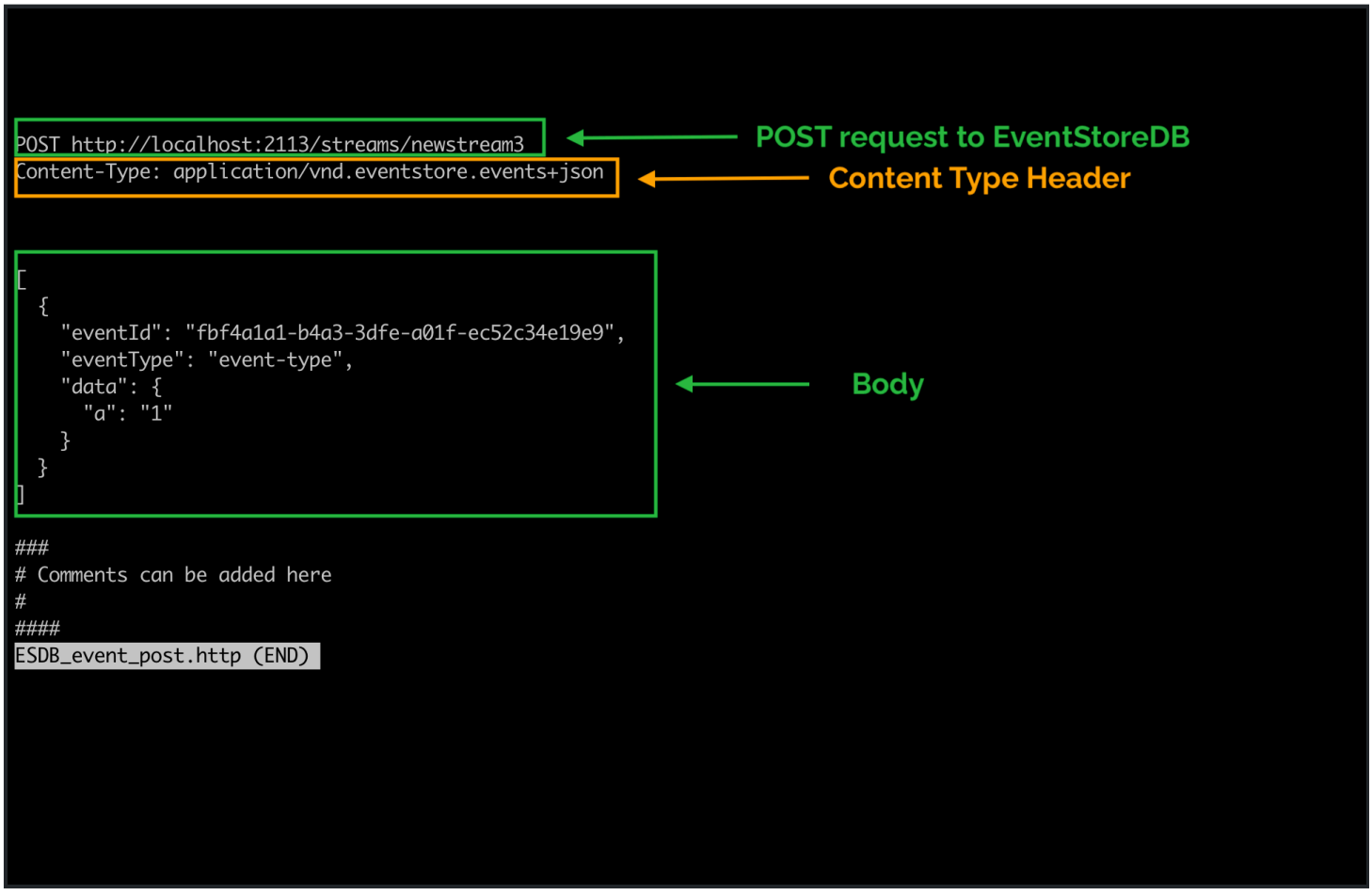
Example:

```
GET https://httpbin.org/anything
Content-Type: text/html
Authorization: Bearer token
```

Request Body

The request body is entered as lines below the Request Headers. Any line that can not be interpreted as a Header is assumed to be the Request Body.

See the image below.



Here is the text of the file.

```
POST http://localhost:2113/streams/newstream
Content-Type: application/vnd.eventstore.events+json
```

```
[
  {
    "eventId": "fbf4a1a3-b4a3-3dfe-a01f-ec52c34e19e9",
    "eventType": "event-type",
    "data": {
      "a": "1"
    }
  }
]
```

5. Install the httpyac plugin

The ".http" files in this directory are formatted to run using the httpyac VScode extension.

A powershell script is included that will install the plugin.

Run

```
.\install_vscode_httpyac_plugin.ps1
```

For information on the plugin see:

[httpyac plugin for vscode]

6. Run some examples

Directories for http requests against a secure and an insecure EventStoreDB database are included.

If there is a `start_cluster.sh` and you are working in windows it can be ignored. That scripts is for starting a docker container in linux, and for this class you will start EventStoreDB directly.

Basic http GET request

The example

```
00_basic_get.http
```

This runs some get requests to Google. The purpose is to show the file format and gain experience running the httpyac httpclient in VScode.

When you run an http request, the response from the server will open in VScode next to the window in which you executed the command.

You may want to close those after each run, or your VScode will get rather crowded.

Please use the "run" command to run one or more http GET requests to google.

Post an event to EventStoreDB

The file,

```
01_ESDB_event_post.http
```

will post an event to the stream "newstream".

Note, if you submit two events with the same eventId in quick succession, the server will drop the second event, detecting that it is a duplicate.

```
POST http://localhost:2113/streams/newstream
Content-Type: application/vnd.eventstore.events+json


[
  {
    "eventId": "fbf4a1a3-b4a3-3dfe-a01f-ec52c34e19e9",
    "eventType": "event-type",
    "data": {
```

```
      "a": "1"
    }
  }
]
```

This will post an event to the stream newstream.

Please use the “send” button to append this event to EventStoreDB.

When you send this request, you should see a `“HTTP/1.1 201 - created”` response in the response window.

You should also be able to see the stream in the stream browser.

Read a single event from EventStoreDB

`02_Read_a_single_event.http`

Sending this request will return a response containing the Event that was appended in the previous example.

Here is the text of the file.

```
GET http://127.0.0.1:2113/streams/newstream/0
Accept:application/vnd.eventstore.atom+json
```

Please use the “send” button in vscode to send the request to the EventStoreDB instance.

The response window will show the content of the event.

7. Comparison of httpyac format to curl

If you prefer cURL commands, here is a quick comparison.

This httpyac format to post an event to a stream

```
POST http://localhost:2113/streams/newstream
Content-Type: application/vnd.eventstore.events+json



[
  {
    "eventId": "fbf4a1a3-b4a3-3dfe-a01f-ec52c34e19e9",
    "eventType": "event-type",
    "data": {
      "a": "1"
    }
  }
]
```

Would be the equivalent to the following cURL command.

```
curl \-i \-d "@event.json" "http://127.0.0.1:2113/streams/newstream3" \\
    \-H "Content-Type:application/vnd.eventstore.events+json"
```

With a data file event.json containing.

```
[
  {
    "eventId": "fbf4a1a3-b4a3-3dfe-a01f-ec52c34e19e9",
    "eventType": "event-type",
    "data": {
      "a": "1"
    }
  }
]
```

Congratulations !!!

You have used the http api to append and read an event from EventStoreDB