

---

# **GNES Documentation**

*Release 0.0.31*

**Han Xiao, Feng Wang, Jianfeng Yan, Jie Fu**

**Aug 20, 2019**



<b>1</b>	<b>Highlights</b>	<b>3</b>
1.1	All Microservices in GNES	3
1.1.1	Named Arguments	3
1.1.2	GNES sub-commands	3
1.1.3	Sub-commands:	4
1.2	gnes package	16
1.2.1	Subpackages	16
1.2.2	Submodules	42
1.2.3	Module contents	43
1.3	Contributing to GNES	43
1.3.1	Table of Content	44
1.3.2	Commit Message Naming	44
1.3.3	Merging Process	44
1.3.4	Release Process	45
1.3.5	Testing Locally	45
1.4	Release Note (v0.0.30)	46
1.4.1	New Features	46
1.4.2	Bug fixes	46
1.4.3	Other Improvements	46
1.5	Release Note (v0.0.29)	46
1.5.1	Bug fixes	46
1.5.2	Code Refactoring	46
1.5.3	Documentation	46
1.5.4	Unit Test and CICD	46
1.5.5	Other Improvements	47
1.6	Release Note (v0.0.28)	47
1.6.1	New Features	47
1.6.2	Bug fixes	47
1.6.3	Documentation	47
1.6.4	Unit Test and CICD	47
1.6.5	Other Improvements	48
1.7	Release Note (v0.0.27)	48
1.7.1	New Features	48
1.7.2	Bug fixes	48
1.7.3	Code Refactoring	49
1.7.4	Unit Test and CICD	49

1.7.5	Other Improvements . . . . .	49
1.8	Release Note (v0.0.26) . . . . .	49
1.8.1	New Features . . . . .	50
1.8.2	Bug fixes . . . . .	50
1.8.3	Code Refactoring . . . . .	50
1.8.4	Documentation . . . . .	50
1.8.5	Other Improvements . . . . .	50
1.9	Release Note (v0.0.25) . . . . .	51
1.9.1	New Features . . . . .	51
1.9.2	Bug fixes . . . . .	51
1.9.3	Code Refactoring . . . . .	52
1.9.4	Documentation . . . . .	52
1.9.5	Unit Test and CICD . . . . .	52
1.9.6	Other Improvements . . . . .	53
1.10	Release Note (v0.0.24) . . . . .	53
1.10.1	New Features . . . . .	53
1.10.2	Bug fixes . . . . .	53
1.10.3	Code Refactoring . . . . .	54
1.10.4	Unit Test and CICD . . . . .	54
1.10.5	Other Improvements . . . . .	54
1.11	Release Note (v0.0.23) . . . . .	54
1.11.1	New Features . . . . .	54
1.11.2	Bug fixes . . . . .	54
1.11.3	Documentation . . . . .	55
1.11.4	Unit Test and CICD . . . . .	55
1.11.5	Other Improvements . . . . .	55
<b>2</b>	<b>Tutorials</b> . . . . .	<b>57</b>
2.1	Troubleshooting . . . . .	57
2.1.1	Check if docker swarm/stack runs successfully . . . . .	57
2.1.2	Investigate the reason of a failed service . . . . .	58
2.1.3	Delete a failed service . . . . .	58
2.1.4	Locate internal errors by looking at logs . . . . .	58
2.2	Protobuf Implementation . . . . .	59
2.2.1	Generating gnes_pb2.py and gnes_pb2_grpc.py . . . . .	59
<b>3</b>	<b>Indices and tables</b> . . . . .	<b>61</b>
	<b>Python Module Index</b> . . . . .	<b>63</b>
	<b>Index</b> . . . . .	<b>65</b>



GNES *jee-nes* is **Generic Neural Elastic Search**, a cloud-native semantic search system based on deep neural network.

GNES enables large-scale index and semantic search for **text-to-text**, **image-to-image**, **video-to-video** and *any-to-any* content form.



## 1.1 All Microservices in GNES

[32mGNES v0.0.31: Generic Neural Elastic Search[0m, a cloud-native semantic search system based on deep neural network. It enables large-scale index and semantic search for text-to-text, image-to-image, video-to-video and any content form. Visit [4m[36m<https://gnes.ai>[0m for tutorials and documentations.

```
usage: gnes [-h] [-v] [--verbose]
           {frontend,encode,index,route,preprocess,grpc,client,compose} ...
```

### 1.1.1 Named Arguments

<b>-v, --version</b>	show program's version number and exit
<b>--verbose</b>	turn on detailed logging for debug
	Default: False

### 1.1.2 GNES sub-commands

use “gnes [sub-command] –help” to get detailed information about each sub-command

<b>cli</b>	Possible choices: frontend, encode, index, route, preprocess, grpc, client, compose
------------	---

### 1.1.3 Sub-commands:

#### frontend

start a frontend service

```
gnes frontend [-h] [--port_in PORT_IN] [--port_out PORT_OUT]
              [--host_in HOST_IN] [--host_out HOST_OUT]
              [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
              [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
              [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
              [--dump_interval DUMP_INTERVAL] [--read_only]
              [--parallel_backend {thread,process}]
              [--num_parallel NUM_PARALLEL]
              [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
              [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
              [--max_message_size MAX_MESSAGE_SIZE]
              [--max_concurrency MAX_CONCURRENCY]
```

#### Named Arguments

<b>--port_in</b>	port for input data, default a random port between [49152, 65536] Default: 59772
<b>--port_out</b>	port for output data, default a random port between [49152, 65536] Default: 49263
<b>--host_in</b>	host address for input Default: "0.0.0.0"
<b>--host_out</b>	host address for output Default: "0.0.0.0"
<b>--socket_in</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
<b>--socket_out</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
<b>--port_ctrl</b>	port for controlling the service, default a random port between [49152, 65536] Default: 60224
<b>--timeout</b>	timeout (ms) of all communication, -1 for waiting forever Default: -1



- dump\_interval** serialize the service to a file every n seconds  
Default: 5
- read\_only** do not allow the service to modify the model, dump\_interval will be ignored  
Default: True
- parallel\_backend** Possible choices: thread, process  
parallel backend of the service  
Default: "thread"
- num\_parallel** number of parallel services running at the same time, *port\_in* and *port\_out* will be set to random, and routers will be added automatically when necessary  
Default: 1
- parallel\_type** Possible choices: PUSH\_BLOCK, PUSH\_NONBLOCK, PUB\_BLOCK, PUB\_NONBLOCK  
parallel type of the concurrent services  
Default: PUSH\_NONBLOCK
- grpc\_host** host address of the grpc service  
Default: "0.0.0.0"
- grpc\_port** host port of the grpc service  
Default: 8800
- max\_message\_size** maximum send and receive size for grpc server in (MB)  
Default: 100
- max\_concurrency** maximum concurrent client allowed  
Default: 10

## encode

start an encoder service

```

gnex encode [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
            [--host_out HOST_OUT]
            [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
            [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
            [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
            [--dump_interval DUMP_INTERVAL] [--read_only]
            [--parallel_backend {thread,process}]
            [--num_parallel NUM_PARALLEL]
            [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
            --yaml_path YAML_PATH [--py_path PY_PATH [PY_PATH ...]]

```

## Named Arguments

- port\_in** port for input data, default a random port between [49152, 65536]

	Default: 60214
<b>--port_out</b>	port for output data, default a random port between [49152, 65536] Default: 62547
<b>--host_in</b>	host address for input Default: "0.0.0.0"
<b>--host_out</b>	host address for output Default: "0.0.0.0"
<b>--socket_in</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
<b>--socket_out</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
<b>--port_ctrl</b>	port for controlling the service, default a random port between [49152, 65536] Default: 63005
<b>--timeout</b>	timeout (ms) of all communication, -1 for waiting forever Default: -1
<b>--dump_interval</b>	serialize the service to a file every n seconds Default: 5
<b>--read_only</b>	do not allow the service to modify the model, dump_interval will be ignored Default: False
<b>--parallel_backend</b>	Possible choices: thread, process parallel backend of the service Default: "thread"
<b>--num_parallel</b>	number of parallel services running at the same time, <i>port_in</i> and <i>port_out</i> will be set to random, and routers will be added automatically when necessary Default: 1
<b>--parallel_type</b>	Possible choices: PUSH_BLOCK, PUSH_NONBLOCK, PUB_BLOCK, PUB_NONBLOCK parallel type of the concurrent services Default: PUSH_NONBLOCK
<b>--yaml_path</b>	yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
<b>--py_path</b>	the file path(s) of an external python module(s).

## index

start an indexer service

```

gnex index [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
           [--host_out HOST_OUT]
           [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
           [--dump_interval DUMP_INTERVAL] [--read_only]
           [--parallel_backend {thread,process}] [--num_parallel NUM_PARALLEL]
           [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
           --yaml_path YAML_PATH [--py_path PY_PATH [PY_PATH ...]]

```

## Named Arguments

<b>--port_in</b>	port for input data, default a random port between [49152, 65536] Default: 58795
<b>--port_out</b>	port for output data, default a random port between [49152, 65536] Default: 58797
<b>--host_in</b>	host address for input Default: "0.0.0.0"
<b>--host_out</b>	host address for output Default: "0.0.0.0"
<b>--socket_in</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_CONNECT
<b>--socket_out</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUB_BIND
<b>--port_ctrl</b>	port for controlling the service, default a random port between [49152, 65536] Default: 55971
<b>--timeout</b>	timeout (ms) of all communication, -1 for waiting forever Default: -1
<b>--dump_interval</b>	serialize the service to a file every n seconds Default: 5

- read\_only** do not allow the service to modify the model, *dump\_interval* will be ignored  
Default: False
- parallel\_backend** Possible choices: thread, process  
parallel backend of the service  
Default: “thread”
- num\_parallel** number of parallel services running at the same time, *port\_in* and *port\_out* will be set to random, and routers will be added automatically when necessary  
Default: 1
- parallel\_type** Possible choices: PUSH\_BLOCK, PUSH\_NONBLOCK, PUB\_BLOCK, PUB\_NONBLOCK  
parallel type of the concurrent services  
Default: PUSH\_NONBLOCK
- yaml\_path** yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
- py\_path** the file path(s) of an external python module(s).

## route

start a router service

```
gnes route [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
           [--host_out HOST_OUT]
           [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
           [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
           [--dump_interval DUMP_INTERVAL] [--read_only]
           [--parallel_backend {thread,process}] [--num_parallel NUM_PARALLEL]
           [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
           --yaml_path YAML_PATH [--py_path PY_PATH [PY_PATH ...]]
           [--num_part NUM_PART]
```

## Named Arguments

- port\_in** port for input data, default a random port between [49152, 65536]  
Default: 56989
- port\_out** port for output data, default a random port between [49152, 65536]  
Default: 49829
- host\_in** host address for input  
Default: “0.0.0.0”
- host\_out** host address for output  
Default: “0.0.0.0”

<b>--socket_in</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
<b>--socket_out</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
<b>--port_ctrl</b>	port for controlling the service, default a random port between [49152, 65536] Default: 63719
<b>--timeout</b>	timeout (ms) of all communication, -1 for waiting forever Default: -1
<b>--dump_interval</b>	serialize the service to a file every n seconds Default: 5
<b>--read_only</b>	do not allow the service to modify the model, dump_interval will be ignored Default: True
<b>--parallel_backend</b>	Possible choices: thread, process parallel backend of the service Default: "thread"
<b>--num_parallel</b>	number of parallel services running at the same time, <i>port_in</i> and <i>port_out</i> will be set to random, and routers will be added automatically when necessary Default: 1
<b>--parallel_type</b>	Possible choices: PUSH_BLOCK, PUSH_NONBLOCK, PUB_BLOCK, PUB_NONBLOCK parallel type of the concurrent services Default: PUSH_NONBLOCK
<b>--yaml_path</b>	yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
<b>--py_path</b>	the file path(s) of an external python module(s).
<b>--num_part</b>	explicitly set the number of parts of message

## preprocess

start a preprocessor service

```

gnex preprocess [-h] [--port_in PORT_IN] [--port_out PORT_OUT]
                [--host_in HOST_IN] [--host_out HOST_OUT]
                [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↪SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]

```

(continues on next page)

(continued from previous page)

```

    [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,
↳ SUB_CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
    [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
    [--dump_interval DUMP_INTERVAL] [--read_only]
    [--parallel_backend {thread,process}]
    [--num_parallel NUM_PARALLEL]
    [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
    --yaml_path YAML_PATH [--py_path PY_PATH [PY_PATH ...]]

```

## Named Arguments

<b>--port_in</b>	port for input data, default a random port between [49152, 65536] Default: 64831
<b>--port_out</b>	port for output data, default a random port between [49152, 65536] Default: 51117
<b>--host_in</b>	host address for input Default: "0.0.0.0"
<b>--host_out</b>	host address for output Default: "0.0.0.0"
<b>--socket_in</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND
<b>--socket_out</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
<b>--port_ctrl</b>	port for controlling the service, default a random port between [49152, 65536] Default: 56900
<b>--timeout</b>	timeout (ms) of all communication, -1 for waiting forever Default: -1
<b>--dump_interval</b>	serialize the service to a file every n seconds Default: 5
<b>--read_only</b>	do not allow the service to modify the model, dump_interval will be ignored Default: True
<b>--parallel_backend</b>	Possible choices: thread, process parallel backend of the service Default: "thread"

<b>--num_parallel</b>	number of parallel services running at the same time, <i>port_in</i> and <i>port_out</i> will be set to random, and routers will be added automatically when necessary Default: 1
<b>--parallel_type</b>	Possible choices: PUSH_BLOCK, PUSH_NONBLOCK, PUB_BLOCK, PUB_NONBLOCK parallel type of the concurrent services Default: PUSH_NONBLOCK
<b>--yaml_path</b>	yaml config of the service, it should be a readable stream, or a valid file path, or a supported class name.
<b>--py_path</b>	the file path(s) of an external python module(s).

## grpc

start a general purpose grpc service

```

gnex grpc [-h] [--port_in PORT_IN] [--port_out PORT_OUT] [--host_in HOST_IN]
          [--host_out HOST_OUT]
          [--socket_in {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
          [--socket_out {PULL_BIND,PULL_CONNECT,PUSH_BIND,PUSH_CONNECT,SUB_BIND,SUB_
↪CONNECT,PUB_BIND,PUB_CONNECT,PAIR_BIND,PAIR_CONNECT}]
          [--port_ctrl PORT_CTRL] [--timeout TIMEOUT]
          [--dump_interval DUMP_INTERVAL] [--read_only]
          [--parallel_backend {thread,process}] [--num_parallel NUM_PARALLEL]
          [--parallel_type {PUSH_BLOCK,PUSH_NONBLOCK,PUB_BLOCK,PUB_NONBLOCK}]
          [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
          [--max_message_size MAX_MESSAGE_SIZE] --pb2_path PB2_PATH
          --pb2_grpc_path PB2_GRPC_PATH --stub_name STUB_NAME --api_name
          API_NAME

```

## Named Arguments

<b>--port_in</b>	port for input data, default a random port between [49152, 65536] Default: 53628
<b>--port_out</b>	port for output data, default a random port between [49152, 65536] Default: 51016
<b>--host_in</b>	host address for input Default: "0.0.0.0"
<b>--host_out</b>	host address for output Default: "0.0.0.0"
<b>--socket_in</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for input port Default: PULL_BIND

<b>--socket_out</b>	Possible choices: PULL_BIND, PULL_CONNECT, PUSH_BIND, PUSH_CONNECT, SUB_BIND, SUB_CONNECT, PUB_BIND, PUB_CONNECT, PAIR_BIND, PAIR_CONNECT socket type for output port Default: PUSH_BIND
<b>--port_ctrl</b>	port for controlling the service, default a random port between [49152, 65536] Default: 64091
<b>--timeout</b>	timeout (ms) of all communication, -1 for waiting forever Default: -1
<b>--dump_interval</b>	serialize the service to a file every n seconds Default: 5
<b>--read_only</b>	do not allow the service to modify the model, dump_interval will be ignored Default: False
<b>--parallel_backend</b>	Possible choices: thread, process parallel backend of the service Default: "thread"
<b>--num_parallel</b>	number of parallel services running at the same time, <i>port_in</i> and <i>port_out</i> will be set to random, and routers will be added automatically when necessary Default: 1
<b>--parallel_type</b>	Possible choices: PUSH_BLOCK, PUSH_NONBLOCK, PUB_BLOCK, PUB_NONBLOCK parallel type of the concurrent services Default: PUSH_NONBLOCK
<b>--grpc_host</b>	host address of the grpc service Default: "0.0.0.0"
<b>--grpc_port</b>	host port of the grpc service Default: 8800
<b>--max_message_size</b>	maximum send and receive size for grpc server in (MB) Default: 100
<b>--pb2_path</b>	the path of the python file protocol buffer compiler
<b>--pb2_grpc_path</b>	the path of the python file generated by the gRPC Python protocol compiler plugin
<b>--stub_name</b>	the name of the gRPC Stub
<b>--api_name</b>	the api name for calling the stub



## client

start a GNES client of the selected type

```
gnес client [-h] {http,cli,benchmark} ...
```

## GNES client sub-commands

use “gnес client [sub-command] –help” to get detailed information about each client sub-command

**client**                    Possible choices: http, cli, benchmark

## Sub-commands:

### http

start a client that allows HTTP requests as input

```
gnес client http [-h] [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
                 [--max_message_size MAX_MESSAGE_SIZE] [--http_port HTTP_PORT]
                 [--http_host HTTP_HOST] [--max_workers MAX_WORKERS]
                 [--top_k TOP_K] [--batch_size BATCH_SIZE]
```

## Named Arguments

<b>--grpc_host</b>	host address of the grpc service Default: “0.0.0.0”
<b>--grpc_port</b>	host port of the grpc service Default: 8800
<b>--max_message_size</b>	maximum send and receive size for grpc server in (MB) Default: 100
<b>--http_port</b>	http port to deploy the service Default: 80
<b>--http_host</b>	http host to deploy the service Default: “0.0.0.0”
<b>--max_workers</b>	max workers to deal with the message Default: 100
<b>--top_k</b>	default top_k for query mode Default: 10
<b>--batch_size</b>	batch size for feed data for train mode Default: 2560

## cli

start a client that allows stdin as input

```
gnes client cli [-h] [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
                [--max_message_size MAX_MESSAGE_SIZE]
                [--txt_file TXT_FILE | --image_zip_file IMAGE_ZIP_FILE | --video_zip_
↵file VIDEO_ZIP_FILE]
                [--batch_size BATCH_SIZE] --mode {index,query,train}
                [--top_k TOP_K] [--start_doc_id START_DOC_ID]
```

## Named Arguments

<b>--grpc_host</b>	host address of the grpc service Default: "0.0.0.0"
<b>--grpc_port</b>	host port of the grpc service Default: 8800
<b>--max_message_size</b>	maximum send and receive size for grpc server in (MB) Default: 100
<b>--txt_file</b>	text file to be used, each line is a doc/query Default: <_io.TextIOWrapper name='<stdin>' mode='r' encoding='UTF-8'>
<b>--image_zip_file</b>	image zip file to be used, consists of multiple images
<b>--video_zip_file</b>	video zip file to be used, consists of multiple videos
<b>--batch_size</b>	the size of the request to split Default: 100
<b>--mode</b>	Possible choices: index, query, train the mode of the client and the server
<b>--top_k</b>	top_k results returned in the query mode Default: 10
<b>--start_doc_id</b>	the start number of doc id Default: 0

## benchmark

start a client for benchmark and unittest

```
gnes client benchmark [-h] [--grpc_host GRPC_HOST] [--grpc_port GRPC_PORT]
                      [--max_message_size MAX_MESSAGE_SIZE]
                      [--batch_size BATCH_SIZE]
                      [--request_length REQUEST_LENGTH]
                      [--num_requests NUM_REQUESTS]
```

## Named Arguments

<b>--grpc_host</b>	host address of the grpc service Default: "0.0.0.0"
<b>--grpc_port</b>	host port of the grpc service Default: 8800
<b>--max_message_size</b>	maximum send and receive size for grpc server in (MB) Default: 100
<b>--batch_size</b>	the size of the request to split Default: 64
<b>--request_length</b>	binary string length of each request Default: 1024
<b>--num_requests</b>	number of total requests Default: 128

## compose

start a GNES Board to visualize YAML configs

```

gnex compose [-h] [--port PORT] [--name NAME] [--yaml_path YAML_PATH]
              [--html_path HTML_PATH] [--shell_path SHELL_PATH]
              [--swarm_path SWARM_PATH] [--k8s_path K8S_PATH]
              [--graph_path GRAPH_PATH]
              [--shell_log_redirect SHELL_LOG_REDIRECT] [--mermaid_leftright]
              [--docker_img DOCKER_IMG] [--flask] [--http_port HTTP_PORT]

```

## Named Arguments

<b>--port</b>	host port of the grpc service Default: 8800
<b>--name</b>	name of the instance Default: "GNES app"
<b>--yaml_path</b>	yaml config of the service Default: <_io.BufferedReader name='/home/docs/checkouts/readthedocs.org/user_builds/gnes/checkouts/example.yml'>
<b>--html_path</b>	output path of the HTML file, will contain all possible generations
<b>--shell_path</b>	output path of the shell-based starting script
<b>--swarm_path</b>	output path of the docker-compose file for Docker Swarm
<b>--k8s_path</b>	output path of the docker-compose file for Docker Swarm
<b>--graph_path</b>	output path of the mermaid graph file

- shell\_log\_redirect** the file path for redirecting shell output. when not given, the output will be flushed to stdout
- mermaid\_leftright** showing the flow in left-to-right manner rather than top down  
Default: False
- docker\_img** the docker image used in Docker Swarm & Kubernetes  
Default: "gnes/gnes:latest-alpine"
- flask** using Flask to serve a composer in interactive mode, aka GNES board  
Default: False
- http\_port** server port for receiving HTTP requests  
Default: 8080

## 1.2 gnes package

### 1.2.1 Subpackages

#### gnes.base package

##### Module contents

**class** `gnes.base.TrainableBase (*args, **kwargs)`

Bases: `object`

The base class for preprocessor, encoder, indexer and router

**close** ()

Release the resources as model is destroyed

**dump** (*filename: str = None*) → None

Serialize the object to a binary file :param filename: file path of the serialized file, if not given then `self.dump_full_path` is used

**dump\_full\_path**

Get the binary dump path

##### Returns

**dump\_yaml** (*filename: str = None*) → None

Serialize the object to a yaml file :param filename: file path of the yaml file, if not given then `self.dump_yaml_path` is used

**classmethod from\_yaml** (*constructor, node, stop\_on\_import\_error=False*)

**static load** (*filename: str = None*) → T

**classmethod load\_yaml** (*filename: Union[str, TextIO]*) → T

**post\_init** ()

Declare class attributes/members that can not be serialized in standard way

**classmethod pre\_init** ()

**store\_args\_kwargs = False**

**classmethod to\_yaml** (*representer, data*)

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**yaml\_full\_path**  
Get the file path of the yaml config :return:

**class** gnes.base.**CompositionalTrainableBase** (\*args, \*\*kwargs)  
Bases: *gnes.base.TrainableBase*

**close** ()  
Release the resources as model is destroyed

**components**

**classmethod** **from\_yaml** (*constructor, node*)

**is\_pipeline**

**classmethod** **to\_yaml** (*representer, data*)

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

## gnes.cli package

### Submodules

#### gnes.cli.api module

gnes.cli.api.**client** (*args*)  
gnes.cli.api.**compose** (*args*)  
gnes.cli.api.**encode** (*args*)  
gnes.cli.api.**frontend** (*args*)  
gnes.cli.api.**grpc** (*args*)  
gnes.cli.api.**index** (*args*)  
gnes.cli.api.**preprocess** (*args*)  
gnes.cli.api.**route** (*args*)

#### gnes.cli.parser module

gnes.cli.parser.**get\_main\_parser** ()  
gnes.cli.parser.**resolve\_py\_path** (*path*)  
gnes.cli.parser.**resolve\_yaml\_path** (*path*)  
gnes.cli.parser.**set\_base\_parser** ()  
gnes.cli.parser.**set\_client\_benchmark\_parser** (*parser=None*)  
gnes.cli.parser.**set\_client\_cli\_parser** (*parser=None*)  
gnes.cli.parser.**set\_client\_http\_parser** (*parser=None*)  
gnes.cli.parser.**set\_composer\_flask\_parser** (*parser=None*)

```
gnes.cli.parser.set_composer_parser (parser=None)
gnes.cli.parser.set_encoder_parser (parser=None)
gnes.cli.parser.set_frontend_parser (parser=None)
gnes.cli.parser.set_grpc_service_parser (parser=None)
gnes.cli.parser.set_indexer_parser (parser=None)
gnes.cli.parser.set_preprocessor_parser (parser=None)
gnes.cli.parser.set_router_parser (parser=None)
gnes.cli.parser.set_service_parser (parser=None)
```

## Module contents

```
gnes.cli.main()
```

## gnes.client package

### Submodules

#### gnes.client.base module

```
class gnes.client.base.ZmqClient (args)
    Bases: object

    close ()

    recv_message (timeout: int = -1) → gnes_pb2.Message

    send_message (message: gnes_pb2.Message, timeout: int = -1)
```

#### gnes.client.benchmark module

```
class gnes.client.benchmark.BenchmarkClient (args)
    Bases: object
```

#### gnes.client.cli module

```
class gnes.client.cli.CLIClient (args)
    Bases: object
```

#### gnes.client.http module

```
class gnes.client.http.HttpClient (args=None)
    Bases: object

    start ()
```

## Module contents

### gnes.composer package

#### Submodules

#### gnes.composer.base module

```

class gnes.composer.base.YamlComposer (args)
    Bases: object

    class Layer (layer_id: int = 0)
        Bases: object

        append (comp)

        default_values = {'income': 'pull', 'name': None, 'replicas': 1, 'yaml_path': M

        get_component_name

        static get_value (comp: Dict, key: str)

        is_heto_single_component

        is_homo_multi_component

        is_homogenous

        is_single_component

        add_comp (comp: Dict) → None

        add_layer (layer: Layer = None) → None

        build_all ()

        static build_dockerswarm (all_layers: List[YamlComposer.Layer], docker_img: str =
            'gnes/gnes:latest-alpine', volumes: Dict = None, networks: Dict =
            None) → str

        static build_html (generate_dict: Dict[str, str]) → str

        static build_kubernetes (all_layers: List[YamlComposer.Layer], *args, **kwargs)

        build_layers () → List[gnes.composer.base.YamlComposer.Layer]

        static build_mermaid (all_layers: List[YamlComposer.Layer], mermaid_leftright: bool = False)
            → str

        static build_shell (all_layers: List[YamlComposer.Layer], log_redirect: str = None) → str

        check_fields (comp: Dict) → bool

        comp2args = {'Encoder': Namespace(dump_interval=5, host_in='0.0.0.0', host_out='0.0.0

        comp2file = {'Encoder': 'encode', 'Frontend': 'frontend', 'Indexer': 'index', 'Prep

```

#### gnes.composer.flask module

```

class gnes.composer.flask.YamlComposerFlask (args)
    Bases: object

    run ()

```

## Module contents

`gnes.encoder` package

Subpackages

`gnes.encoder.audio` package

Submodules

`gnes.encoder.audio.mfcc` module

```
class gnes.encoder.audio.mfcc.MfccEncoder (n_mfcc: int = 13, sample_rate: int = 16000,  
                                           max_length: int = 100, *args, **kwargs)  
    Bases: gnes.encoder.base.BaseAudioEncoder  
  
    batch_size = 64  
  
    encode (data: List[np.array], *args, **kwargs) → numpy.ndarray  
  
    train (*args, **kwargs)  
        Train the model, need to be overridden
```

## Module contents

`gnes.encoder.image` package

Subpackages

`gnes.encoder.image.cvae_cores` package

Submodules

`gnes.encoder.image.cvae_cores.model` module

## Module contents

`gnes.encoder.image.inception_cores` package

Submodules

`gnes.encoder.image.inception_cores.inception_utils` module

`gnes.encoder.image.inception_cores.inception_v4` module

## Module contents

Submodules



### gnes.encoder.image.base module

```
class gnes.encoder.image.base.BasePytorchEncoder (model_name: str, layers: List[str],
                                                model_dir: str, use_cuda: bool =
                                                False, *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

**batch\_size** = 64

**encode** (*img: List[np.ndarray], \*args, \*\*kwargs*) → numpy.ndarray

**post\_init** ()

Declare class attributes/members that can not be serialized in standard way

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

### gnes.encoder.image.cvae module

```
class gnes.encoder.image.cvae.CVAEEncoder (model_dir: str, latent_dim: int = 300, se-
                                            lect_method: str = 'MEAN', l2_normalize:
                                            bool = False, use_gpu: bool = True, *args,
                                            **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

**batch\_size** = 64

**encode** (*img: List[np.ndarray], \*args, \*\*kwargs*) → numpy.ndarray

**post\_init** ()

Declare class attributes/members that can not be serialized in standard way

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

### gnes.encoder.image.inception module

```
class gnes.encoder.image.inception.TFInceptionEncoder (model_dir: str, select_layer:
                                                         str = 'PreLogitsFlatten',
                                                         use_cuda: bool = False,
                                                         *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

**batch\_size** = 64

**encode** (*img: List[np.ndarray], \*args, \*\*kwargs*) → numpy.ndarray

**post\_init** ()

Declare class attributes/members that can not be serialized in standard way

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

## gnes.encoder.image.onnx module

```
class gnes.encoder.image.onnx.BaseONNXImageEncoder (model_name: str, model_dir: str,  
use_cuda: bool = False, *args,  
**kwargs)
```

Bases: *gnes.encoder.base.BaseImageEncoder*

```
batch_size = 64
```

```
encode (img: List[np.ndarray], *args, **kwargs) → numpy.ndarray
```

```
post_init ()
```

Declare class attributes/members that can not be serialized in standard way

```
train (*args, **kwargs)
```

Train the model, need to be overridden

## Module contents

### gnes.encoder.numeric package

#### Submodules

### gnes.encoder.numeric.hash module

```
class gnes.encoder.numeric.hash.HashEncoder (num_bytes: int, num_bits: int = 8, num_idx:  
int = 3, kmeans_clusters: int = 100, method:  
str = 'product_uniform', *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseNumericEncoder*

```
batch_size = 2048
```

```
encode (vecs: numpy.ndarray, *args, **kwargs) → numpy.ndarray
```

```
hash (vecs)
```

```
pred_kmeans (vecs)
```

```
ran_gen ()
```

```
train (vecs: numpy.ndarray, *args, **kwargs)
```

Train the model, need to be overridden

```
train_kmeans (vecs)
```

### gnes.encoder.numeric.pca module

```
class gnes.encoder.numeric.pca.PCALocalEncoder (output_dim: int, num_locals: int, *args,  
**kwargs)
```

Bases: *gnes.encoder.base.BaseNumericEncoder*

```
batch_size = 2048
```

```
encode (vecs: numpy.ndarray, *args, **kwargs) → numpy.ndarray
```

```
train (vecs: numpy.ndarray, *args, **kwargs) → None
```

Train the model, need to be overridden

## gnes.encoder.numeric.pq module

```
class gnes.encoder.numeric.pq.PQEncoder (num_bytes: int, cluster_per_byte: int = 255, *args,
                                         **kwargs)
    Bases: gnes.encoder.base.BaseBinaryEncoder

    batch_size = 2048

    encode (vecs: numpy.ndarray, *args, **kwargs) → numpy.ndarray

    train (vecs: numpy.ndarray, *args, **kwargs)
        Train the model, need to be overridden
```

## gnes.encoder.numeric.tf\_pq module

```
class gnes.encoder.numeric.tf_pq.TFPQEncoder (num_bytes: int, cluster_per_byte: int =
                                                255, *args, **kwargs)
    Bases: gnes.encoder.numeric.pq.PQEncoder

    batch_size = 8192

    close ()
        Release the resources as model is destroyed

    encode (vecs: numpy.ndarray, *args, **kwargs) → numpy.ndarray

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    classmethod pre_init ()

    train (vecs: numpy.ndarray, *args, **kwargs)
```

## gnes.encoder.numeric.vlad module

```
class gnes.encoder.numeric.vlad.VladEncoder (num_clusters: int, *args, **kwargs)
    Bases: gnes.encoder.base.BaseNumericEncoder

    batch_size = 2048

    encode (vecs: numpy.ndarray, *args, **kwargs) → numpy.ndarray

    kmeans_pred (vecs)

    kmeans_train (vecs)

    train (vecs: numpy.ndarray, *args, **kwargs)
        Train the model, need to be overridden
```

## Module contents

### gnes.encoder.text package

### Submodules

### gnes.encoder.text.bert module

```
class gnes.encoder.text.bert.BertEncoder (*args, **kwargs)
    Bases: gnes.encoder.base.BaseTextEncoder

    close ()
        Release the resources as model is destroyed

    encode (text: List[str], *args, **kwargs) → numpy.ndarray

    is_trained = True

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    store_args_kwargs = True

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.encoder.text.bert.BertEncoderServer (*args, **kwargs)
    Bases: gnes.encoder.base.BaseTextEncoder

    close ()
        Release the resources as model is destroyed

    is_trained = True

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    store_args_kwargs = True

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.encoder.text.bert.BertEncoderWithServer (*args, **kwargs)
    Bases: gnes.base.CompositionalTrainableBase

    encode (text: List[str], *args, **kwargs) → numpy.ndarray

    train (*args, **kwargs)
        Train the model, need to be overridden
```

### gnes.encoder.text.elmo module

```
class gnes.encoder.text.elmo.ElmoEncoder (model_dir: str, pooling_layer: int = -1, pooling_strategy: str = 'REDUCE_MEAN', *args, **kwargs)
    Bases: gnes.encoder.base.BaseTextEncoder

    batch_size = 64

    encode (text: List[str], *args, **kwargs) → numpy.ndarray

    is_trained = True

    post_init ()
        Declare class attributes/members that can not be serialized in standard way

    train (*args, **kwargs)
        Train the model, need to be overridden
```

### gnes.encoder.text.flair module

```
class gnes.encoder.text.flair.FlairEncoder (model_name: str = 'multi-forward-fast', pooling_strategy: str = 'REDUCE_MEAN', *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseTextEncoder*

```
encode (text: List[str], *args, **kwargs) → numpy.ndarray
```

```
is_trained = True
```

```
post_init ()
```

Declare class attributes/members that can not be serialized in standard way

```
train (*args, **kwargs)
```

Train the model, need to be overridden

### gnes.encoder.text.gpt module

```
class gnes.encoder.text.gpt.GPT2Encoder (model_dir: str, use_cuda: bool = False, pooling_strategy: str = 'REDUCE_MEAN', *args, **kwargs)
```

Bases: *gnes.encoder.text.gpt.GPTEncoder*

```
train (*args, **kwargs)
```

Train the model, need to be overridden

```
class gnes.encoder.text.gpt.GPTEncoder (model_dir: str, use_cuda: bool = False, pooling_strategy: str = 'REDUCE_MEAN', *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseTextEncoder*

```
batch_size = 64
```

```
encode (text: List[str], *args, **kwargs) → numpy.ndarray
```

```
is_trained = True
```

```
post_init ()
```

Declare class attributes/members that can not be serialized in standard way

```
train (*args, **kwargs)
```

Train the model, need to be overridden

### gnes.encoder.text.torch\_transformers module

```
class gnes.encoder.text.torch_transformers.TorchTransformersEncoder (model_dir: str, model_name: str, tokenizer_name: str, use_cuda: bool = False, *args, **kwargs)
```

Bases: *gnes.encoder.base.BaseTextEncoder*

```
encode (text: List[str], *args, **kwargs) → numpy.ndarray  
is_trained = True  
post_init ()  
    Declare class attributes/members that can not be serialized in standard way  
train (*args, **kwargs)  
    Train the model, need to be overridden
```

### **gnes.encoder.text.w2v module**

```
class gnes.encoder.text.w2v.Word2VecEncoder (model_dir: str, skiprows: int = 1, dimension: int = 300, pooling_strategy: str = 'REDUCE_MEAN', *args, **kwargs)  
  
    Bases: gnes.encoder.base.BaseTextEncoder  
  
    encode (text: List[str], *args, **kwargs) → numpy.ndarray  
  
    is_trained = True  
  
    post_init ()  
        Declare class attributes/members that can not be serialized in standard way  
  
    train (*args, **kwargs)  
        Train the model, need to be overridden
```

### **Module contents**

#### **gnes.encoder.video package**

#### **Subpackages**

#### **gnes.encoder.video.mixture\_core package**

#### **Submodules**

#### **gnes.encoder.video.mixture\_core.model module**

#### **Module contents**

#### **Submodules**

## gnes.encoder.video.incep\_mixture module

```
class gnes.encoder.video.incep_mixture.IncepMixtureEncoder (model_dir_inception:
                                                    str,
                                                    model_dir_mixture:
                                                    str, select_layer: str
                                                    = 'PreLogitsFlatten',
                                                    use_cuda: bool =
                                                    False, feature_size:
                                                    int = 300, vocab_size:
                                                    int = 28, cluster_size:
                                                    int = 256, method: str
                                                    = 'fvnet', input_size:
                                                    int = 1536, vo-
                                                    cab_size_2: int = 174,
                                                    max_frames: int = 30,
                                                    multitask_method: str
                                                    = 'Attention', *args,
                                                    **kwargs)
```

Bases: *gnes.encoder.base.BaseVideoEncoder*

**batch\_size** = 64

**encode** (*data: List[np.ndarray], \*args, \*\*kwargs*) → numpy.ndarray

**post\_init** ()

Declare class attributes/members that can not be serialized in standard way

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

## Module contents

### Submodules

#### gnes.encoder.base module

```
class gnes.encoder.base.BaseAudioEncoder (*args, **kwargs)
```

Bases: *gnes.encoder.base.BaseEncoder*

**encode** (*data: List[np.ndarray], \*args, \*\*kwargs*) → numpy.ndarray

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

```
class gnes.encoder.base.BaseBinaryEncoder (*args, **kwargs)
```

Bases: *gnes.encoder.base.BaseEncoder*

**encode** (*data: numpy.ndarray, \*args, \*\*kwargs*) → bytes

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

```
class gnes.encoder.base.BaseEncoder (*args, **kwargs)
```

Bases: *gnes.base.TrainableBase*

**encode** (*data: Any, \*args, \*\*kwargs*) → Any

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.encoder.base.**BaseImageEncoder** (\*args, \*\*kwargs)  
Bases: *gnes.encoder.base.BaseEncoder*

**encode** (img: List[np.ndarray], \*args, \*\*kwargs) → numpy.ndarray

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.encoder.base.**BaseNumericEncoder** (\*args, \*\*kwargs)  
Bases: *gnes.encoder.base.BaseEncoder*

**encode** (text: numpy.ndarray, \*args, \*\*kwargs) → numpy.ndarray

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.encoder.base.**BaseTextEncoder** (\*args, \*\*kwargs)  
Bases: *gnes.encoder.base.BaseEncoder*

**encode** (text: List[str], \*args, \*\*kwargs) → numpy.ndarray

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.encoder.base.**BaseVideoEncoder** (\*args, \*\*kwargs)  
Bases: *gnes.encoder.base.BaseEncoder*

**encode** (data: List[np.ndarray], \*args, \*\*kwargs) → numpy.ndarray

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.encoder.base.**PipelineEncoder** (\*args, \*\*kwargs)  
Bases: *gnes.base.CompositionalTrainableBase*

**encode** (data: Any, \*args, \*\*kwargs) → Any

**train** (data, \*args, \*\*kwargs)  
Train the model, need to be overridden

## Module contents

**gnes.indexer package**

**Subpackages**

**gnes.indexer.fulltext package**

**Submodules**



## gnes.indexer.fulltext.leveldb module

```
class gnes.indexer.fulltext.leveldb.AsyncLVDBIndexer (data_path: str,
keep_na_doc: bool = True,
drop_raw_bytes: bool = False,
drop_chunk_blob: bool = False, *args, **kwargs)
```

Bases: *gnes.indexer.fulltext.leveldb.LVDBIndexer*

```
add (keys: List[int], docs: List[gnes_pb2.Document], *args, **kwargs)
```

```
close ()
    Release the resources as model is destroyed
```

```
post_init ()
    Declare class attributes/members that can not be serialized in standard way
```

```
query (*args, **kwargs) → List[Any]
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

```
class gnes.indexer.fulltext.leveldb.LVDBIndexer (data_path: str, keep_na_doc: bool =
True, drop_raw_bytes: bool = False,
drop_chunk_blob: bool = False, *args,
**kwargs)
```

Bases: *gnes.indexer.base.BaseTextIndexer*

```
add (keys: List[int], docs: List[gnes_pb2.Document], *args, **kwargs)
```

```
close ()
    Release the resources as model is destroyed
```

```
post_init ()
    Declare class attributes/members that can not be serialized in standard way
```

```
query (keys: List[int], *args, **kwargs) → List[gnes_pb2.Document]
```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

## Module contents

### gnes.indexer.vector package

#### Subpackages

### gnes.indexer.vector.bindexer package

#### Submodules

### gnes.indexer.vector.bindexer.bindexer module

## Module contents

## gnes.indexer.vector.hbindexer package

### Submodules

#### gnes.indexer.vector.hbindexer.hbindexer module

### Module contents

### Submodules

#### gnes.indexer.vector.annoy module

```
class gnes.indexer.vector.annoy.AnnoyIndexer (num_dim: int, data_path: str, metric: str =
                                             'angular', n_trees=10, *args, **kwargs)
    Bases: gnes.indexer.base.BaseVectorIndexer
    add (keys: List[Tuple[int, Any]], vectors: numpy.ndarray, weights: List[float], *args, **kwargs)
    normalize_score (score: List[float], metrics: str, *args, **kwargs) → List[float]
    post_init ()
        Declare class attributes/members that can not be serialized in standard way
    query ()
    size
    train (*args, **kwargs)
        Train the model, need to be overridden
```

#### gnes.indexer.vector.faiss module

```
class gnes.indexer.vector.faiss.FaissIndexer (num_dim: int, index_key: str, data_path:
                                             str, *args, **kwargs)
    Bases: gnes.indexer.base.BaseVectorIndexer
    add (keys: List[Tuple[int, Any]], vectors: numpy.ndarray, weights: List[float], *args, **kwargs)
    normalize_score (score: numpy.ndarray, *args, **kwargs) → numpy.ndarray
    post_init ()
        Declare class attributes/members that can not be serialized in standard way
    query ()
    size
    train (*args, **kwargs)
        Train the model, need to be overridden
```

#### gnes.indexer.vector.numpy module

```
class gnes.indexer.vector.numpy.NumpyIndexer (num_bytes: int = None, *args, **kwargs)
    Bases: gnes.indexer.base.BaseVectorIndexer
    add (keys: List[Tuple[int, Any]], vectors: numpy.ndarray, weights: List[float], *args, **kwargs)
```

```

query ()
train (*args, **kwargs)
    Train the model, need to be overridden

```

## Module contents

### Submodules

#### gnes.indexer.base module

```

class gnes.indexer.base.BaseIndexer (*args, **kwargs)
    Bases: gnes.base.TrainableBase

    add (keys: Any, docs: Any, weights: List[float], *args, **kwargs)
    normalize_score (*args, **kwargs)
    query (keys: Any, *args, **kwargs) → List[Any]
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.base.BaseKeyIndexer (*args, **kwargs)
    Bases: gnes.indexer.base.BaseIndexer

    add (keys: List[Tuple[int, int]], weights: List[float], *args, **kwargs) → int
    query (keys: List[int], *args, **kwargs) → List[Tuple[int, int, float]]
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.base.BaseTextIndexer (*args, **kwargs)
    Bases: gnes.indexer.base.BaseIndexer

    add (keys: List[int], docs: Any, weights: List[float], *args, **kwargs)
    query (keys: List[int], *args, **kwargs) → List[Any]
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.base.BaseVectorIndexer (*args, **kwargs)
    Bases: gnes.indexer.base.BaseIndexer

    add (keys: List[Tuple[int, int]], vectors: numpy.ndarray, weights: List[float], *args, **kwargs)
    query ()
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.base.JointIndexer (*args, **kwargs)
    Bases: gnes.base.CompositionalTrainableBase

    add (keys: Any, docs: Any, *args, **kwargs) → None

    components
    query ()

```

```
train (*args, **kwargs)
    Train the model, need to be overridden
```

## gnex.indexer.key\_only module

```
class gnes.indexer.key_only.DictKeyIndexer (*args, **kwargs)
    Bases: gnex.indexer.base.BaseKeyIndexer

    add (keys: List[Tuple[int, int]], weights: List[float], *args, **kwargs) → int
    query (keys: List[int], *args, **kwargs) → List[Tuple[int, int, float]]
    size

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.key_only.ListKeyIndexer (*args, **kwargs)
    Bases: gnex.indexer.base.BaseKeyIndexer

    add (keys: List[Tuple[int, int]], weights: List[float], *args, **kwargs) → int
    query (keys: List[int], *args, **kwargs) → List[Tuple[int, int, float]]
    size

    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.key_only.ListNumpyKeyIndexer (*args, **kwargs)
    Bases: gnex.indexer.key_only.ListKeyIndexer

    add (*args, **kwargs) → int
    query (keys: List[int], *args, **kwargs) → List[Tuple[int, int, float]]
    train (*args, **kwargs)
        Train the model, need to be overridden

class gnes.indexer.key_only.NumpyKeyIndexer (buffer_size: int = 10000, col_size: int = 3,
                                             *args, **kwargs)
    Bases: gnex.indexer.base.BaseKeyIndexer

    add (keys: List[Tuple[int, int]], weights: List[float], *args, **kwargs) → int
    capacity
    query (keys: List[int], *args, **kwargs) → List[Tuple[int, int, float]]
    size

    train (*args, **kwargs)
        Train the model, need to be overridden
```

## Module contents

### gnex.preprocessor package

### Subpackages

## gnes.preprocessor.audio package

### Submodules

#### gnes.preprocessor.audio.audio\_vanilla module

#### gnes.preprocessor.audio.base module

```
class gnes.preprocessor.audio.base.BaseAudioPreprocessor (uniform_doc_weight:
bool = True, *args,
**kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

**doc\_type** = 4

**train** (*\*args, \*\*kwargs*)  
Train the model, need to be overridden

### Module contents

## gnes.preprocessor.image package

### Submodules

#### gnes.preprocessor.image.base module

```
class gnes.preprocessor.image.base.BaseImagePreprocessor (target_img_size: int =
224, is_rgb: bool = True,
*args, **kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

**doc\_type** = 2

**train** (*\*args, \*\*kwargs*)  
Train the model, need to be overridden

#### gnes.preprocessor.image.resize module

```
class gnes.preprocessor.image.resize.ResizeChunkPreprocessor (target_width: int =
224, target_height:
int = 224, *args,
**kwargs)
```

Bases: *gnes.preprocessor.image.base.BaseImagePreprocessor*

**apply** (*doc: gnes\_pb2.Document*) → None

**train** (*\*args, \*\*kwargs*)  
Train the model, need to be overridden

## gnes.preprocessor.image.segmentation module

```
class gnes.preprocessor.image.segmentation.SegmentPreprocessor (model_name:  
                                                    str, model_dir:  
                                                    str, _use_cuda:  
                                                    bool =  
                                                    False, *args,  
                                                    **kwargs)
```

Bases: *gnes.preprocessor.image.base.BaseImagePreprocessor*

**apply** (*doc: gnes\_pb2.Document*)

**post\_init** ()

Declare class attributes/members that can not be serialized in standard way

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

## gnes.preprocessor.image.sliding\_window module

### Module contents

## gnes.preprocessor.text package

### Submodules

## gnes.preprocessor.text.base module

```
class gnes.preprocessor.text.base.BaseTextPreprocessor (uniform_doc_weight: bool  
                                                    = True, *args, **kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

**doc\_type** = 1

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

## gnes.preprocessor.text.simple module

```
class gnes.preprocessor.text.simple.TextPreprocessor (delimiter: str = '[!?!]+',  
                                                    *args, **kwargs)
```

Bases: *gnes.preprocessor.text.base.BaseTextPreprocessor*

**apply** (*doc: gnes\_pb2.Document*) → None

**train** (*\*args, \*\*kwargs*)

Train the model, need to be overridden

### Module contents

## gnes.preprocessor.video package

## Submodules

### gnes.preprocessor.video.base module

```
class gnes.preprocessor.video.base.BaseVideoPreprocessor (uniform_doc_weight:
                                                    bool = True, *args,
                                                    **kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

**doc\_type** = 3

**train** (*\*args, \*\*kwargs*)  
Train the model, need to be overridden

### gnes.preprocessor.video ffmpeg module

### gnes.preprocessor.video.shotdetect module

## Module contents

## Submodules

### gnes.preprocessor.base module

```
class gnes.preprocessor.base.BasePreprocessor (uniform_doc_weight: bool = True, *args,
                                                    **kwargs)
```

Bases: *gnes.base.TrainableBase*

**apply** (*doc: gnes\_pb2.Document*) → None

**doc\_type** = 0

**train** (*\*args, \*\*kwargs*)  
Train the model, need to be overridden

```
class gnes.preprocessor.base.PipelinePreprocessor (*args, **kwargs)
```

Bases: *gnes.base.CompositionalTrainableBase*

**apply** (*doc: gnes\_pb2.Document*) → None

**train** (*data, \*args, \*\*kwargs*)  
Train the model, need to be overridden

```
class gnes.preprocessor.base.UnaryPreprocessor (doc_type: int, *args, **kwargs)
```

Bases: *gnes.preprocessor.base.BasePreprocessor*

**apply** (*doc: gnes\_pb2.Document*)

**is\_trained** = True

**raw\_to\_chunk** (*chunk: gnes\_pb2.Chunk, raw\_bytes: bytes*)

**train** (*\*args, \*\*kwargs*)  
Train the model, need to be overridden

## gnes.preprocessor.helper module

### Module contents

## gnes.proto package

### Submodules

## gnes.proto.gnes\_pb2 module

## gnes.proto.gnes\_pb2\_grpc module

**class** gnes.proto.gnes\_pb2\_grpc.GnesRPCServicer

Bases: object

**Call** (*request, context*)

**Index** (*request, context*)

**Query** (*request, context*)

**StreamCall** (*request\_iterator, context*)

**Train** (*request, context*)

option (rpc\_core.method\_no\_deadline) = true; option (rpc\_core.service\_default\_deadline\_ms) = 5000;

**class** gnes.proto.gnes\_pb2\_grpc.GnesRPCStub (*channel*)

Bases: object

Constructor.

**Args:** channel: A grpc.Channel.

gnes.proto.gnes\_pb2\_grpc.add\_GnesRPCServicer\_to\_server (*servicer, server*)

### Module contents

**class** gnes.proto.RequestGenerator

Bases: object

**static index** (*data: List[bytes], start\_doc\_id: int = 0, start\_request\_id: int = 0, random\_doc\_id: bool = False, batch\_size: int = 0, doc\_type: int = 1, \*args, \*\*kwargs*)

**static query** (*query: bytes, top\_k: int, start\_request\_id: int = 0, doc\_type: int = 1, \*args, \*\*kwargs*)

**static train** (*data: List[bytes], start\_doc\_id: int = 0, start\_request\_id: int = 0, random\_doc\_id: bool = False, batch\_size: int = 0, doc\_type: int = 1, \*args, \*\*kwargs*)

gnes.proto.send\_message (*sock: zmq.sugar.socket.Socket, msg: gnes\_pb2.Message, timeout: int = -1*)  
→ None

gnes.proto.recv\_message (*sock: zmq.sugar.socket.Socket, timeout: int = -1*) → Optional[*gnes\_pb2.Message*]

gnes.proto.blob2array (*blob: gnes\_pb2.NdArray*) → numpy.ndarray  
Convert a blob proto to an array.

gnes.proto.array2blob (*x: numpy.ndarray*) → gnes\_pb2.NdArray  
Converts a N-dimensional array to blob proto.

gnes.proto.add\_route (*evlp: gnes\_pb2.Envelope, name: str*)



## gnes.router package

### Subpackages

## gnes.router.map package

### Submodules

## gnes.router.map.simple module

**class** `gnes.router.map.simple.DocBatchRouter` (\*args, \*\*kwargs)

Bases: `gnes.router.base.BaseMapRouter`

**apply** (`msg: gnes_pb2.Message`, \*args, \*\*kwargs) → Generator  
Modify the incoming message

**Parameters** `msg` – incoming message

**train** (\*args, \*\*kwargs)

Train the model, need to be overridden

**class** `gnes.router.map.simple.PublishRouter` (`num_part: int`, \*args, \*\*kwargs)

Bases: `gnes.router.base.BaseMapRouter`

**apply** (`msg: gnes_pb2.Message`, \*args, \*\*kwargs) → Generator  
Modify the incoming message

**Parameters** `msg` – incoming message

**train** (\*args, \*\*kwargs)

Train the model, need to be overridden

### Module contents

## gnes.router.reduce package

### Submodules

## gnes.router.reduce.chunk module

**class** `gnes.router.reduce.chunk.ChunkToDocumentRouter` (\*args, \*\*kwargs)

Bases: `gnes.router.base.BaseReduceRouter`

**apply** (`msg: gnes_pb2.Message`, `accum_msgs: List[gnes_pb2.Message]`, \*args, \*\*kwargs)  
Modify the current message based on accumulated messages

#### Parameters

- `msg` – the current message
- `accum_msgs` – accumulated messages

**train** (\*args, \*\*kwargs)

Train the model, need to be overridden

### gnes.router.reduce.chunk\_sum module

```
class gnes.router.reduce.chunk_sum.ChunkSumRouter (*args, **kwargs)
    Bases: gnes.router.base.BaseReduceRouter

apply (msg: gnes_pb2.Message, accum_msgs: List[gnes_pb2.Message], *args, **kwargs)
    Modify the current message based on accumulated messages

    Parameters

    • msg – the current message

    • accum_msgs – accumulated messages

train (*args, **kwargs)
    Train the model, need to be overridden
```

### gnes.router.reduce.concat module

```
class gnes.router.reduce.concat.ConcatEmbedRouter (*args, **kwargs)
    Bases: gnes.router.base.BaseReduceRouter

apply (msg: gnes_pb2.Message, accum_msgs: List[gnes_pb2.Message], *args, **kwargs)
    Modify the current message based on accumulated messages

    Parameters

    • msg – the current message

    • accum_msgs – accumulated messages

train (*args, **kwargs)
    Train the model, need to be overridden
```

### gnes.router.reduce.doc\_sum module

```
class gnes.router.reduce.doc_sum.DocSumRouter (*args, **kwargs)
    Bases: gnes.router.base.BaseReduceRouter

apply (msg: gnes_pb2.Message, accum_msgs: List[gnes_pb2.Message], *args, **kwargs)
    Modify the current message based on accumulated messages

    Parameters

    • msg – the current message

    • accum_msgs – accumulated messages

train (*args, **kwargs)
    Train the model, need to be overridden
```

### gnes.router.reduce.document module

```
class gnes.router.reduce.document.DocFillRouter (*args, **kwargs)
    Bases: gnes.router.base.BaseReduceRouter

apply (msg: gnes_pb2.Message, accum_msgs: List[gnes_pb2.Message], *args, **kwargs)
    Modify the current message based on accumulated messages
```

**Parameters**

- **msg** – the current message
- **accum\_msgs** – accumulated messages

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**Module contents****Submodules****gnes.router.base module**

**class** gnes.router.base.**BaseMapRouter** (\*args, \*\*kwargs)  
Bases: *gnes.router.base.BaseRouter*

**apply** (msg: *gnes\_pb2.Message*, \*args, \*\*kwargs) → Generator  
Modify the incoming message

**Parameters** **msg** – incoming message

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.router.base.**BaseReduceRouter** (\*args, \*\*kwargs)  
Bases: *gnes.router.base.BaseRouter*

**apply** (msg: *gnes\_pb2.Message*, accum\_msgs: *List[gnes\_pb2.Message]*, \*args, \*\*kwargs) → None  
Modify the current message based on accumulated messages

**Parameters**

- **msg** – the current message
- **accum\_msgs** – accumulated messages

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

**class** gnes.router.base.**BaseRouter** (\*args, \*\*kwargs)  
Bases: *gnes.base.TrainableBase*

Base class for the router. Inherit from this class to create a new router.

Router forwards messages between services. Essentially, it receives a 'gnes\_pb2.Message' and call *apply()* method on it.

**apply** (msg: *gnes\_pb2.Message*, \*args, \*\*kwargs)  
Modify the incoming message

**Parameters** **msg** – incoming message

**train** (\*args, \*\*kwargs)  
Train the model, need to be overridden

## Module contents

### gnes.service package

#### Submodules

#### gnes.service.base module

```
class gnes.service.base.BaseService (args)
    Bases: object

    close ()

    default_host = '0.0.0.0'

    dump ()

    handler = <gnes.service.base.MessageHandler object>

    load_model (base_class: Type[gnes.base.TrainableBase]) → T

    message_handler (msg: gnes_pb2.Message, out_sck, ctrl_sck)

    post_init ()

    run ()

    status

class gnes.service.base.BetterEnum
    Bases: enum.Enum

    An enumeration.

    from_string = <bound method BetterEnum.from_string of <enum 'BetterEnum'>>

exception gnes.service.base.BlockMessage
    Bases: Exception

exception gnes.service.base.ComponentNotLoad
    Bases: Exception

class gnes.service.base.ConcurrentService
    Bases: type

exception gnes.service.base.EventLoopEnd
    Bases: Exception

class gnes.service.base.MessageHandler (mh: Optional[gnes.service.base.MessageHandler] = None)
    Bases: object

    register ()

    serve (msg: gnes_pb2.Message)

class gnes.service.base.ParallelType
    Bases: gnes.service.base.BetterEnum

    An enumeration.

    PUB_BLOCK = 2

    PUB_NONBLOCK = 3
```

```

    PUSH_BLOCK = 0
    PUSH_NONBLOCK = 1
    is_block
    is_push
class gnes.service.base.ReduceOp
    Bases: gnes.service.base.BetterEnum
    An enumeration.
    ALWAYS_ONE = 1
    CONCAT = 0
exception gnes.service.base.ServiceError
    Bases: Exception
class gnes.service.base.ServiceManager(service_cls, args)
    Bases: object
    join()
class gnes.service.base.SocketType
    Bases: gnes.service.base.BetterEnum
    An enumeration.
    PAIR_BIND = 8
    PAIR_CONNECT = 9
    PUB_BIND = 6
    PUB_CONNECT = 7
    PULL_BIND = 0
    PULL_CONNECT = 1
    PUSH_BIND = 2
    PUSH_CONNECT = 3
    SUB_BIND = 4
    SUB_CONNECT = 5
    is_bind
gnes.service.base.build_socket(ctx: zmq.sugar.context.Context, host: str, port: int, socket_type:
    gnes.service.base.SocketType, identity: Optional[str] = None)
    → Tuple[zmq.sugar.socket.Socket, str]
gnes.service.base.send_ctrl_message(address: str, msg: gnes_pb2.Message, timeout: int)

```

### gnes.service.encoder module

```

class gnes.service.encoder.EncoderService(args)
    Bases: gnes.service.base.BaseService
    static get_chunks_from_docs(docs: Union[List[gnes_pb2.Document], gnes_pb2.Document])
        → List
    handler = <gnes.service.base.MessageHandler object>

```

```
post_init()
```

### gnes.service.frontend module

```
class gnes.service.frontend.FrontendService(args)
    Bases: object
```

### gnes.service.grpc module

```
class gnes.service.grpc.GRPCService(args)
    Bases: gnes.service.base.BaseService

    close()
    handler = <gnes.service.base.MessageHandler object>
    post_init()
```

### gnes.service.indexer module

```
class gnes.service.indexer.IndexerService(args)
    Bases: gnes.service.base.BaseService

    handler = <gnes.service.base.MessageHandler object>
    post_init()
```

### gnes.service.preprocessor module

```
class gnes.service.preprocessor.PreprocessorService(args)
    Bases: gnes.service.base.BaseService

    handler = <gnes.service.base.MessageHandler object>
    post_init()
```

### gnes.service.router module

```
class gnes.service.router.RouterService(args)
    Bases: gnes.service.base.BaseService

    handler = <gnes.service.base.MessageHandler object>
    post_init()
```

## Module contents

### 1.2.2 Submodules

#### gnes.helper module

```
gnes.helper.get_sys_info()
```

```

gnes.helper.get_optimal_sample_size(x)
gnes.helper.get_perm(L, m)
gnes.helper.time_profile(func)
gnes.helper.set_logger(context, verbose=False)
gnes.helper.batch_iterator(data: Union[Iterator[Any], List[Any], numpy.ndarray], batch_size:
                           int, axis: int = 0) → Iterator[Any]
gnes.helper.batching(func: Callable[Any, numpy.ndarray] = None, *, batch_size: Union[int,
                                           Callable] = None, num_batch=None, iter_axis: int = 0, concat_axis: int =
                                           0, chunk_dim=-1)
gnes.helper.load_contrib_module()
gnes.helper.parse_arg(v: str)
gnes.helper.profiling(func)
class gnes.helper.FileLock(lock_file: str = 'LOCK')
    Bases: object
    Implements the Posix based file locking (Linux, Ubuntu, MacOS, etc.)
    acquire()
    is_locked
    release()
gnes.helper.train_required(func)
gnes.helper.get_first_available_gpu()
class gnes.helper.PathImporter
    Bases: object
    static add_modules(*paths)

```

### 1.2.3 Module contents

## 1.3 Contributing to GNES

Thanks for your interest in contributing! GNES always welcome the contribution from the open-source community, individual committers and other partners. Without you, GNES can't be successful.

Currently there are three major directions of contribution:

- **Porting state-of-the-art models to GNES.** This includes new preprocessing algorithms, new DNN networks for encoding, and new high-performance index. Believe me, it is super easy to wrap an algorithm and use it in GNES. Checkout this example.
- **Adding tutorial and learning experience.** What is good and what can be improved? If you apply GNES in your domain, whether it's about NLP or CV, whether it's a blog post or a Reddit/Twitter thread, we are always eager to hear your thoughts.
- **Completing the user experience of other programming languages.** GNES offers a generic interface with gRPC and protobuf, therefore it is easy to add an interface for other languages, e.g. Java, C, Go.

### 1.3.1 Table of Content

- *Commit Message Naming*
- *Merging Process*
- *Release Process*
  - *Major and minor version increments*
- *Testing Locally*

### 1.3.2 Commit Message Naming

To help everyone with understanding the commit history of GNES, we employ `commitlint` in the CI pipeline to enforce the commit styles. Specifically, our convention is:

```
type(scope?): subject
```

where `type` is one of the following:

- build
- ci
- chore
- docs
- feat
- fix
- perf
- refactor
- revert
- style
- test

`scope` is optional, represents the module your commit working on.

`subject` explains the commit.

As an example, a commit that implements a new encoder should be phrased as:

```
feat(encoder): add new inceptionV3 as image encoder
```

### 1.3.3 Merging Process

A pull request has to meet the following conditions to be merged into master:

- Coding style check (PEP8, via Codacy)
- Commit style check (in CI pipeline via Drone.io)
- Unit tests (via Drone.io)
- Review and approval from a GNES team member.

After the merging is triggered, the build will be delivered to the followings:



- **Docker Hub:** `gnes:latest` will be updated.
- **Tencent Container Service:** `gnes:latest` will be updated.
- **ReadTheDoc:** `latest` will be updated.

Note that merging into master does not mean an official releasing. For the releasing process, please refer to the next section.

### 1.3.4 Release Process

A new release is scheduled on every Friday (triggered and approved by Han Xiao) summarizing all new commits since the last release. The release will increment the third (revision) part of the version number, i.e. from `0.0.24` to `0.0.25`.

After a release is triggered, the build will be delivered to the followings:

- **Docker Hub:** a new image with the release version tag will be created, `gnes:latest` will be updated.
- **Tencent Container Service:** a new image with the release version tag will be created, `gnes:latest` will be updated.
- **PyPi Package:** a new version of Python package is uploaded to Pypi, allowing one to `pip install -U gnes`
- **ReadTheDoc:** a new version of the document will be built, `latest` will be updated and the old version will be achieved.

Meanwhile, a new pull request containing the updated **CHANGELOG** and the new version number will be made automatically, pending for review and merge.

### Major and minor version increments

- MAJOR version when GNES make incompatible API changes;
- MINOR version when GNES add functionality in a backwards-compatible manner.

The decision of incrementing major and minor version, i.e. from `0.0.0` to `0.1.0` or from `1.0.0` to `2.0.0`, is made by the GNES team.

### 1.3.5 Testing Locally

The best way to test GNES is using a Docker container, in which you don't have to worry about the dependencies.

We provide a public Docker image `gnes/ci-base`, which contains the required dependencies and some pretrained models used in our continuous integration pipeline.

You can [find the image at here](#) or pull the image via:

```
docker pull gnes/ci-base
```

To test GNES inside this image, you may run

```
docker run --network=host --rm --entrypoint "/bin/bash" -it gnes/ci-base
# now you are inside the 'gnes/ci-base' container
# first sync your local modification, then
pip install -e .[all]
python -m unittest tests/*.py
```

## 1.4 Release Note (v0.0.30)

Release time: 2019-08-19 14:13:03

We'd like to thank all contributors for this new release! In particular, hanhxiao,

### 1.4.1 New Features

- [7b5cc86a] - **contrib**: no need to give module name in advance (*hanhxiao*)

### 1.4.2 Bug fixes

- [5f69c781] - **contrib**: allowing dump for contributed module (*hanhxiao*)

### 1.4.3 Other Improvements

- [565ef569] - **changelog**: update change log to v0.0.29 (*hanhxiao*)

## 1.5 Release Note (v0.0.29)

Release time: 2019-08-16 15:40:31

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem,

### 1.5.1 Bug fixes

- [2f905168] - **setup**: fix encoding problem in setup.py (*hanhxiao*)
- [469bc51d] - **ci**: fix cd pipeline (*hanhxiao*)

### 1.5.2 Code Refactoring

- [66d020bd] - **base**: component renamed to components (*hanhxiao*)
- [3a2b85b6] - **proto**: assign doc id in request generator (*Jem*)

### 1.5.3 Documentation

- [b854c697] - **readme**: fix description on images (*hanhxiao*)

### 1.5.4 Unit Test and CICD

- [f1658c92] - **docker**: docker image tag-alpine as default tag (*hanhxiao*)
- [8885512f] - **docker**: clean up the space after docker build (*hanhxiao*)

## 1.5.5 Other Improvements

- [00ca6919] - **changelog**: update change log to v0.0.28 (*hanhxiao*)

## 1.6 Release Note (v0.0.28)

Release time: 2019-08-14 20:54:26

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, raccoonliukai, Larry Yan,

### 1.6.1 New Features

- [0133905c] - **client**: add a client for benchmarking and testing (*hanhxiao*)
- [732f2e64] - **encoder**: add pytorch transformers support in text encoder (*raccoonliukai*)
- [6aab48c8] - **docker**: add buster image with minimum dependencies (*hanhxiao*)
- [da1bbc0d] - **docker**: add alpine image with minimum dependencies (*hanhxiao*)

### 1.6.2 Bug fixes

- [315bd16a] - **doc sum router**: use meta info instead of doc id to do doc sum (*Jem*)
- [c9e92722] - **encoder**: use offline model in ci-base for pytorch transformer (*raccoonliukai*)
- [d7b42d39] - **setup**: remove unused dependencies (*hanhxiao*)
- [5b8acf7c] - **test**: fix routes assert in tests (*hanhxiao*)
- [5fedf6df] - **encoder**: fix unused variable (*raccoonliukai*)
- [df616463] - **cli**: remove unnecessary argument (*hanhxiao*)
- [fd76aa79] - **request\_generator**: send index request in index mode (*Jem*)
- [64163cb1] - **batching**: enable to process three dimension output in batching (*Jem*)
- [415456d6] - **preprocessor**: fix bug (*Larry Yan*)
- [c150ad59] - **preprocessor**: modify ffmpeg video pre add video cutting method (*Larry Yan*)
- [b0f22d04] - **audio preprocessor**: filter audio with zero length (*Jem*)
- [d1cfa539] - **preprocessor**: modify ffmpeg video preprocessor (*Larry Yan*)

### 1.6.3 Documentation

- [e11a920e] - **readme**: add image explain to readme (*hanhxiao*)

### 1.6.4 Unit Test and CICD

- [a8700801] - **drone**: add self hosted drone (*hanhxiao*)
- [079d0a1a] - **docker**: move docker-build to a more controllable cd process (*hanhxiao*)

## 1.6.5 Other Improvements

- [5257259f] - add kai liu to core maintainers (*hanhxiao*)
- [8d318204] - **changelog**: update change log to v0.0.27 (*hanhxiao*)

## 1.7 Release Note (v0.0.27)

Release time: 2019-08-09 19:51:57

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, Larry Yan, raccoonliu, Han Xiao, raccoonliukai,

### 1.7.1 New Features

- [55126f2b] - **grpc**: add a general purpose grpc service (*hanhxiao*)
- [23c6e68a] - **reduce router**: add chunk and doc reduce routers for audio (*Jem*)
- [6d3d2b4c] - **cli**: use ServiceManager as default service runner (*hanhxiao*)
- [ccfd474a] - **service**: add ServiceManager and enable parallel services in one container (*hanhxiao*)
- [63f9173f] - **service**: enabling the choose of thread or process as the backend (*hanhxiao*)
- [2647b848] - **audio**: add preprocess and mfcc encoder for audio (*Jem*)
- [208e1937] - **audio**: add preprocess and mfcc encoder for audio, update protobuf (*Jem*)
- [77a2ea42] - **parser**: improve yaml\_path parsing (*hanhxiao*)
- [762535ca] - **vlad**: add vlad and enable multiple chunks and frames (*Jem*)
- [64e948d4] - **encoder**: add onnxruntime for image encoder (*raccoonliukai*)
- [f03e6fc2] - **encoder**: add onnxruntime suport for image encoder (*raccoonliukai*)

### 1.7.2 Bug fixes

- [5ae46d61] - **composer**: rename grpcfrontend to frontend (*hanhxiao*)
- [4cb83383] - **audio**: restrict max length for mfcc encoding (*Jem*)
- [e516646f] - **grpc**: add max\_message\_size to the argparser (*hanhxiao*)
- [0493e6fc] - **encoder**: fix netvlad (*Larry Yan*)
- [e773aa33] - **service manager**: fix nonetype for service manager (*Jem*)
- [d5d15d7f] - **compose**: fix a bug in doc\_reduce\_test (*hanhxiao*)
- [6856cb0a] - **compose**: copy args on every request (*hanhxiao*)
- [f80e8c03] - **cli**: set default num\_part is None (*hanhxiao*)
- [7031fe20] - **preprocessor**: add random sampling to ffmpeg (*Larry Yan*)
- [fd37e6d9] - **encoder**: fix bug caused by batching in inception\_mixture (*Larry Yan*)
- [2191b27b] - **composer**: fix yaml generation (*hanhxiao*)
- [e5fefcee] - **encoder**: fix batching in encoder (*hanhxiao*)

- [e35e3b3c] - **composer**: fix composer router generation logic (*hanhxiao*)
- [7300e055] - **preprocessor**: quantity improvement (*Larry Yan*)
- [47efaba4] - **unittest**: fix unittest of video preprocessor 2 (*Larry Yan*)
- [a6efb4af] - **unittest**: fix unittest of video preprocessor (*Larry Yan*)
- [dd1216bb] - **unittest**: fix unittest for video processor (*Larry Yan*)
- [8e6dc4c6] - **encoder**: add func for preprocessor (*Larry Yan*)
- [2b21dc5a] - **encoder**: fix unused import and variable (*raccoonliu*)
- [fd576915] - **test**: fix import (*Han Xiao*)
- [a0fdad36] - **test**: fix broken code (*Han Xiao*)
- [8ca07a74] - **test**: fix img\_process\_for\_test (*Han Xiao*)
- [7c16fb8b] - **preprocessor**: fix bug in ffmpeg.py and add more func to helper (*Larry Yan*)
- [e6a37119] - **preprocessor**: fix bug in params in ffmpeg (*Larry Yan*)
- [f8d2abe5] - **preprocessor**: fix bug in ffmpeg (*Larry Yan*)
- [67610f86] - **preprocessor**: add more method for cutting video (*Larry Yan*)

### 1.7.3 Code Refactoring

- [8516096d] - **grpc**: moving zmqclient to client module (*hanhxiao*)
- [5e3409e1] - **grpc**: hide private class inside gRPCfrontend (*hanhxiao*)
- [6407cc8d] - **yaml**: remove useless default yaml config (*hanhxiao*)
- [c1e406ae] - **onnx**: move batch\_size to class attribute (*Han Xiao*)

### 1.7.4 Unit Test and CICD

- [5503dbe7] - skip joint indexer test as it is not even used (*hanhxiao*)
- [8ab101ca] - add mergify for auto merging (*hanhxiao*)
- [203d1697] - **chore**: exclude chore job from ci pipeline (*hanhxiao*)
- [24f9fd1c] - fix yaml\_path missing in the test (*hanhxiao*)
- [23a83a40] - simplify yaml naming (*hanhxiao*)

### 1.7.5 Other Improvements

- [e8e3b9b9] - **changelog**: update change log to v0.0.26 (*hanhxiao*)

## 1.8 Release Note (v0.0.26)

Release time: 2019-08-02 18:18:45

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, Larry Yan,

### 1.8.1 New Features

- [d0b2ef0b] - **composer**: more interaction for gnes board (*hanhxiao*)
- [9c33dc66] - **router**: allow consecutive mapping and reducing ops (*hanhxiao*)

### 1.8.2 Bug fixes

- [fc5026da] - **board**: improve gnes board 500 message (*hanhxiao*)
- [823bdeda] - **test**: fix grpc gentle shutdown (*hanhxiao*)
- [f6a801f7] - **test**: fix preprocessor building for image test (*hanhxiao*)
- [50fdc041] - **base**: fix ref to CompositionalTrainableBase (*hanhxiao*)
- [54a931c7] - **test**: fix test images by removing mac stuff (*hanhxiao*)
- [14cdfabe] - **sliding window**: fix the boundary (*Jem*)
- [46b5c94e] - **encoder**: fix name for video encoder (*Larry Yan*)
- [15eb50b4] - **encoder**: fix params in basevideo encoder (*Larry Yan*)
- [5b0fe7c6] - **preprocessor**: fix FFmpegVideoSegmentor (*Larry Yan*)
- [d6a46fa6] - **encoder**: fix import path for mixture encoder (*Larry Yan*)
- [17779676] - **encoder**: fix mixture encoder (*Larry Yan*)
- [95f03c56] - **encoder**: fix bug in video mixture encoder (*Larry Yan*)
- [3fd1c06] - **encoder**: fix mixture (*Larry Yan*)
- [67991533] - **encoder**: add netvlad and netfv register class (*Larry Yan*)
- [92500f0f] - **encoder**: add netvlad and netfv (*Larry Yan*)

### 1.8.3 Code Refactoring

- [c430ef64] - **base**: better batch\_size control (*hanhxiao*)
- [58217d8c] - **base**: moving is\_trained to class attribute (*hanhxiao*)
- [7126d496] - **preprocessor**: separate resize logic from the unary preprocessor (*hanhxiao*)
- [52f87c7f] - **base**: make pipelineencoder more general and allow pipelinepreprocessor (*hanhxiao*)

### 1.8.4 Documentation

- [3ab3723e] - **tutorial**: fix image and code layout (*hanhxiao*)

### 1.8.5 Other Improvements

- [635ba37f] - **changelog**: update change log to v0.0.25 (*hanhxiao*)

## 1.9 Release Note (v0.0.25)

Release time: 2019-07-26 19:45:21

We'd like to thank all contributors for this new release! In particular, hanhxiao, felix, Larry Yan, Jem, Han Xiao, Felix,

### 1.9.1 New Features

- [66aec9c9] - **grpc**: add StreamCall and decouple send and receive (*hanhxiao*)
- [5697441b] - **indexer**: consider offset relevance at query time (*Jem*)
- [04c9c745] - **image preprocessor**: calculate offsetnd for each chunk (*Jem*)
- [b34a765a] - **compose**: add interactive mode of GNES board using Flask (*hanhxiao*)
- [5876c15e] - **base**: support loading external modules from py and yaml (*hanhxiao*)

### 1.9.2 Bug fixes

- [a20672d3] - **preprocessor**: add logging in helper module (*felix*)
- [f9500c1f] - **protobuffer**: add doc\_type as func argument in RequestGenerator (*felix*)
- [1c3bb01a] - **service**: fix bug in doc\_type name in indexer service (*Larry Yan*)
- [d834f578] - **service**: add doc type to req generator (*Larry Yan*)
- [80e234e1] - **service**: fix bug in req Generator add doc\_type (*Larry Yan*)
- [5743e258] - **indexer**: fix bug in indexer service (*Larry Yan*)
- [11dde2bf] - **encoder**: fix bug in tf inception (*Larry Yan*)
- [ded92c57] - **indexer**: fix bug for indexer service dealing with empty doc (*Larry Yan*)
- [1dff06f1] - **encoder**: fix bug for encoder service dealing with empty doc (*Larry Yan*)
- [7e43d5a2] - **preprocessor**: fix ffmpeg to deal with broken image (*Larry Yan*)
- [83ebaced] - **preprocessor**: move import imagehash to inside (*hanhxiao*)
- [7c669a70] - **test**: rename the yaml test file (*hanhxiao*)
- [2cc26342] - **compose**: change textarea font to monospace (*hanhxiao*)
- [e644e391] - **encoder**: fix gpu limitation in inception (*Larry Yan*)
- [89d8b70c] - **grpc**: fix bug in RequestGenerator query (*Larry Yan*)
- [c52c2cc6] - **base**: fix gnes\_config mixed in kwargs (*hanhxiao*)
- [68c15fac] - **base**: fix redundant warning in pipeline encoder (*hanhxiao*)
- [aadeeeefb] - **composer**: fix composer state machine (*hanhxiao*)
- [c0bffe6c] - **indexer**: normalize weight (*Jem*)
- [2c696483] - **indexer**: fix weight in indexer call (*Larry Yan*)
- [139a02d9] - **compose**: fix compose bug of pub-sub rule, duplicate yaml\_path (*hanhxiao*)
- [649ed131] - **encoder**: add normalize option in cvae encoder (*Larry Yan*)

- [eb487799] - **encoder**: fix tf scope error in cvae encoder (*Larry Yan*)
- [ab6c88cc] - **encoder**: fix error in cvae encoder (*Larry Yan*)
- [a4b883ac] - **indexer**: add drop raw bytes option to leveldb (*Larry Yan*)
- [4b52bcba] - **grpc**: fix grpc plugin path (*Larry Yan*)
- [d3fbbcac] - **weighting**: add simple normalization to chunk search (*Jem*)
- [08a9a4e3] - **grpc**: fix grpc service (*Larry Yan*)
- [6e6bbf83] - **grpc**: add auto-gen grpc code (*Larry Yan*)
- [b89d8fa2] - **grpc**: add stream index and train in proto (*Larry Yan*)
- [15cd7e58] - **base**: fix dump and load on compositional encoder (*hanhxiao*)
- [bab48919] - **encoder**: fix tf inception (*Larry Yan*)
- [973672ef] - **encoder**: fix bug for encoder bin load (*Larry Yan*)
- [1bef3971] - **setup**: fix setup script (*hanhxiao*)
- [67fb5766] - **compose**: fix argparser (*hanhxiao*)
- [63c4515f] - **compose**: accept parser argument only (*hanhxiao*)
- [887d89cc] - **release**: ask BOT\_URL before releasing (*hanhxiao*)

### 1.9.3 Code Refactoring

- [9973f600] - **preprocessor**: rename singleton to unary (*hanhxiao*)
- [a1a2b020] - **proto**: refactor request stream call (*hanhxiao*)

### 1.9.4 Documentation

- [c853e3da] - **tutorial**: fix svg size (*hanhxiao*)
- [04cccdcd] - **tutorial**: fix svg path (*hanhxiao*)
- [8927cd4f] - **tutorial**: add yaml explain (*hanhxiao*)
- [5b52ce4c] - fix doc path (*hanhxiao*)
- [45751e1f] - **readme**: add quick start for readme (*hanhxiao*)
- [73891ecc] - **readme**: add install guide to readme and contribution guide (*hanhxiao*)

### 1.9.5 Unit Test and CICD

- [6ff3079b] - **unittest**: skip all os environ test (*hanhxiao*)
- [816fa043] - **unittest**: skip blocked test (*hanhxiao*)
- [79a9c106] - **unittest**: run test in verbose mode (*hanhxiao*)
- [83276f90] - **torchvision**: install torchvision dependency to enable tests (*hanhxiao*)
- [499682ce] - **base**: add unit test for load a dumped pipeline from yaml (*hanhxiao*)
- [26a7ad18] - **composer**: add unit test for flask (*hanhxiao*)



- [87ec1fd2] - **base**: move module delete to teardown (*hanhxiao*)
- [479b183d] - **compose**: skip unit test (*hanhxiao*)

## 1.9.6 Other Improvements

- [c30f39cc] - ... (*felix*)
- [2d5654c0] - **license**: add license (*hanhxiao*)
- [d3347910] - reformat code and optimize import (*hanhxiao*)
- [71491ffb] - **changelog**: update change log to v0.0.24 (*hanhxiao*)

## 1.10 Release Note (v0.0.24)

Release time: 2019-07-19 18:18:46

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, Larry Yan, felix,

### 1.10.1 New Features

- [9f6c0524] - **fasterrcnn**: add the original image to chunk list (*Jem*)
- [abb0841c] - **encoder**: add convolution variational autoencoder (*Larry Yan*)

### 1.10.2 Bug fixes

- [1b526832] - **base**: fix dump yaml kwargs (*hanhxiao*)
- [086f3cea] - **base**: fix ump instance (*hanhxiao*)
- [12dfde42] - **base**: move name setting to trainable base (*hanhxiao*)
- [16f1a497] - **base**: move set config to metaclass (*hanhxiao*)
- [b97acd6c] - **base**: fix duplicate warning (*hanhxiao*)
- [991e4425] - **base**: fix duplicate load and init from yaml (*hanhxiao*)
- [69a486e5] - **compose**: fix import (*hanhxiao*)
- [4977aa3c] - **vector indexer**: reorder relevance and chunk weight (*Jem*)
- [2448411d] - **encoder**: modify CVAE (*Larry Yan*)
- [b4bf0bf8] - **indexer**: add path check for dir and file (*hanhxiao*)
- [92f36c33] - **fasterrcnn**: handle imgs with 0 chunk (*Jem*)
- [a1329913] - **fasterrcnn**: fix bug for gpu (*Jem*)
- [38eca0ce] - **grpc**: change grpc client message size limit (*felix*)
- [3836020a] - **preprocessor**: fix preprocessor service handler function name error (*felix*)
- [599a3c3d] - **compose**: fix composer logic (*hanhxiao*)
- [7f3b2fb5] - **release**: fix git tag version (*hanhxiao*)

### 1.10.3 Code Refactoring

- [9bbb3c05] - **compose**: move compose template to resources (*hanhxiao*)
- [a4e153d7] - **base**: remove dump path and reorganize work dir (*hanhxiao*)

### 1.10.4 Unit Test and CICD

- [e088ea9c] - **drone**: turn off profiling in ci (*hanhxiao*)
- [33a570b9] - **drone**: remove pylint for faster ci (*hanhxiao*)
- [51eafac7] - **indexer**: fix data path in unit test (*hanhxiao*)

### 1.10.5 Other Improvements

- [43ef4108] - **git**: add tmp to ignore (*hanhxiao*)
- [44b1a0c9] - fix unittest (*felix*)
- [984a9a2d] - **changelog**: update change log to v0.0.23 (*hanhxiao*)

## 1.11 Release Note (v0.0.23)

Release time: 2019-07-17 18:28:08

We'd like to thank all contributors for this new release! In particular, hanhxiao, Jem, felix, Larry Yan, Han Xiao,

### 1.11.1 New Features

- [cb4d9cf2] - **release**: add auto release and keep change log (*hanhxiao*)
- [c667d874] - **image\_preprocessor**: add fasterRCNN (*Jem*)
- [a6c2975b] - **composer**: improve the gnes board with cards (*hanhxiao*)
- [6ec4233d] - **composer**: add swarm and bash generator (*hanhxiao*)
- [08aa30f4] - **composer**: add shell script generator (*hanhxiao*)
- [033a4b9c] - **composer**: add composer and mermaid renderer (*hanhxiao*)

### 1.11.2 Bug fixes

- [2b7c3f18] - **compose**: resolve unclosed file warning (*hanhxiao*)
- [8030feb2] - **compose**: fix router logic in compose (*hanhxiao*)
- [736f6053] - **gnesboard**: fix cdn (*hanhxiao*)
- [fb07ff02] - **doc\_reducer\_router**: fix reduce error (*felix*)
- [a7236308] - **image\_encoder**: define use\_cuda variable via args (*felix*)
- [cba5e190] - **image\_encoder**: enable batching encoding (*felix*)
- [3423ec83] - **composer**: add compose api to api.py (*hanhxiao*)

- [70ba3fca] - **composer**: in bash mode always run job in background (*hanhxiao*)
- [054981ce] - **composer**: fix gnes board naming (*hanhxiao*)
- [743ec3b0] - **composer**: fix unit test and add tear down (*hanhxiao*)
- [64aef413] - **composer**: fix styling according to codacy (*hanhxiao*)
- [dca4b03b] - **service**: fix bug grpc (*Larry Yan*)
- [09e68da2] - **service**: fix grpc server size limit (*Larry Yan*)
- [3da8da19] - **encoder**: rm un-used import in inception (*Larry Yan*)
- [8780a4da] - bugs for integrated test (*Jem*)
- [38fff782] - **preprocessor**: move cv2 dep to pic\_weight (*Han Xiao*)
- [37155bba] - **preprocessor-video**: move sklearn dep to apply (*Han Xiao*)
- [1f6a06a2] - **encoder**: rm tf inception unittest (*Larry Yan*)
- [eaffbbff] - **encoder**: register tf inception in **init** (*Larry Yan*)
- [d0099b79] - **encoder**: add necessary code from tf (*Larry Yan*)
- [b480774a] - **encoder**: add inception tf (*Larry Yan*)

### 1.11.3 Documentation

- [54276c6a] - **readme**: improve readme image and structure (*hanhxiao*)

### 1.11.4 Unit Test and CI/CD

- [1dcfdfa7] - **docker-image**: optimize docker file (*felix*)
- [bda562d1] - **drone**: auto release with cron job (*hanhxiao*)

### 1.11.5 Other Improvements

- [0c737a94] - **release**: revert back master check (*hanhxiao*)
- [7b04697c] - **changelog**: revert the change log to empty (*hanhxiao*)
- [d02f320d] - revert docker file (*felix*)



**Warning:** Tutorial is still under construction. Stay tuned! Meanwhile, we sincerely welcome you to contribute your own learning experience / case study with GNES!

## 2.1 Troubleshooting

### 2.1.1 Check if docker swarm/stack runs successfully

```
docker service ls
```

ID	NAME	MODE	REPLICAS	
↪	IMAGE	PORTS		
j7b533zxmzg5	gnes-swarm-2654_encoder	replicated	0/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master			
0vlxu4acg1ph	gnes-swarm-2654_income-proxy	replicated	0/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master	*:4962->4962/tcp		
equqrhsn7pky	gnes-swarm-2654_indexer	replicated	0/3	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master			
nd7euo7mcpa9	gnes-swarm-2654_middleman-proxy	replicated	0/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master			
ssdlk9gzmggw	gnes-swarm-2654_outgoing-proxy	replicated	0/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:master	*:4963->4963/tcp		
xgxeetyhos6t	my-gnes_encoder	replicated	1/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f			
zny37400p225	my-gnes_income-proxy	replicated	1/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f	*:8598->8598/tcp		
taqqg6qwrxlw	my-gnes_indexer	replicated	3/3	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f			
j96gnyy8ysbn	my-gnes_middleman-proxy	replicated	1/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f			
e28spnuksjw8	my-gnes_outgoing-proxy	replicated	1/1	↪
↪	ccr.ccs.tencentyun.com/gnes/aipd-gnes:a799a0f	*:8599->8599/tcp		

(continues on next page)

In the above example, we started two swarms, i.e. `gnes-swarm-2654` and `my-gnes`. Unfortunately, `gnes-swarm-2654` fails to start and is not running at all. But how can one tell that?

Note the column `REPLICAS`, which indicates the number of running service (versus the number of required services). `gnes-swarm-2654` gives `0/0` for all services. This suggests the swarm fails to start. The next step is to investigate the reason.

### 2.1.2 Investigate the reason of a failed service

One can not print out all logs of a docker swarm. Instead, one can inspect service by service, e.g.

```
docker service ps gnes-swarm-2654_encoder --format "{{json .Error}}" --no-trunc
```

```
"\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
\"invalid mount config for type \"bind\": bind source path does not exist: /data/han/
↪test-shell/output_data\""
```

Now the reason is clear, `output_data` does not exist when starting the swarm. But why there are duplicated lines there? This is because docker swarm did three retries before giving up on starting this service, where each time it met the same problem. Thus four duplicated lines in total.

### 2.1.3 Delete a failed service

Now that the reason is clear, we can delete the failed service and release the resources.

```
docker stack rm gnes-swarm-2654
```

```
Removing service gnes-swarm-2654_encoder
Removing service gnes-swarm-2654_income-proxy
Removing service gnes-swarm-2654_indexer
Removing service gnes-swarm-2654_middleman-proxy
Removing service gnes-swarm-2654_outgoing-proxy
Removing network gnes-swarm-2654_gnes-net
```

### 2.1.4 Locate internal errors by looking at logs

Sometime the service fails to start but `docker service ps` gives no error,

```
docker service ps gnes-swarm-4254_encoder --format "{{json .Error}}" --no-trunc
```

```
" "
```

Or it shows an error that is not explanatory.

```
"\"task: non-zero exit (2)\\""
```

Often in this case, the service fails to start *not* due to the docker config, but due to the GNES internal error. To see that,

```
docker service logs gnes-swarm-4254_income-proxy
```

```
gnes-swarm-4254_income-proxy.1.yj5v8n4dhfgv@VM-0-3-ubuntu | | [--
↪proxy_type {BS,Dict,MapProxyService,Message,MessageHandler,ProxyService,
↪ReduceProxyService,defaultdict}}
gnes-swarm-4254_income-proxy.1.yj5v8n4dhfgv@VM-0-3-ubuntu | | [--
↪batch_size BATCH_SIZE] [--num_part NUM_PART]
gnes-swarm-4254_income-proxy.1.kmgk21qo6m0n@VM-0-3-ubuntu | | [--
↪proxy_type {BS,Dict,MapProxyService,Message,MessageHandler,ProxyService,
↪ReduceProxyService,defaultdict}}
gnes-swarm-4254_income-proxy.1.w04d552cuj93@VM-0-3-ubuntu | gnes proxy: error:
↪argument --batch_size: invalid int value: ''
gnes-swarm-4254_income-proxy.1.kmgk21qo6m0n@VM-0-3-ubuntu | | [--
↪batch_size BATCH_SIZE] [--num_part NUM_PART]
```

One can now clearly see that the error comes from an incorrectly given `--batch_size`, which throws from GNES CLI.

## 2.2 Protobuf Implementation

The file `gnes/proto/gnes.proto` defines the protobuf used in GNES. It is the core message protocol used in communicating between services. It also defines the interface of a gRPC service.

`gnes_pb2.py` and `gnes_pb2_grpc.py` are python interfaces automatically generated by protobuf tools.

For developers who want to change the protobuf definition, one needs to first edit `gnes/proto/gnes.proto` and then regenerate the python codes (i.e. `gnes_pb2.py` and `gnes_pb2_grpc.py`).

### 2.2.1 Generating `gnes_pb2.py` and `gnes_pb2_grpc.py`

Take MacOS as an example,

1. Download `protoc-$VERSION-$PLATFORM.zip` from [the official site](#) and decompress it.
2. Copy the binary file and include to your system path:

```
cp ~/Downloads/protoc-3.7.1-osx-x86_64/bin/protoc /usr/local/bin/
cp -r ~/Downloads/protoc-3.7.1-osx-x86_64/include/* /usr/local/include/
```

1. Install gRPC tools dependencies: `brew install automake autoconf libtool`
2. Install gRPC and `grpc_python_plugin` from the source:

```
git clone https://github.com/grpc/grpc.git
git submodule update --init
make grpc_python_plugin
```

1. This will compile the `grpc-python-plugin` and build it to, e.g., `/Documents/grpc/bins/opt/grpc_python_plugin`
2. Generate the python codes:

```
SRC_DIR=gnес/proto/  
PLUGIN_PATH=/Documents/grpc/bins/opt/grpc_python_plugin  
  
protoc -I $SRC_DIR --python_out=$SRC_DIR --grpc_python_out=$SRC_DIR --plugin=protoc-  
↳gen-grpc_python=${PLUGIN_PATH} ${SRC_DIR}/gnес.proto
```

1. Fixing the import in `gnес_pb2_grpc.py`. For some reason (probably a bug of gRPC?), the generated code of import is not correct in `gnес_pb2_grpc.py`, you have to change it to the following:

```
# Generated by the gRPC Python protocol compiler plugin. DO NOT EDIT!  
import grpc  
  
from . import gnес_pb2 as gnес__pb2
```



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### g

- gnes, 43
- gnes.base, 16
- gnes.cli, 18
- gnes.cli.api, 17
- gnes.cli.parser, 17
- gnes.client, 19
- gnes.client.base, 18
- gnes.client.benchmark, 18
- gnes.client.cli, 18
- gnes.client.http, 18
- gnes.composer, 20
- gnes.composer.base, 19
- gnes.composer.flask, 19
- gnes.encoder, 28
- gnes.encoder.audio, 20
- gnes.encoder.audio.mfcc, 20
- gnes.encoder.base, 27
- gnes.encoder.image, 22
- gnes.encoder.image.base, 21
- gnes.encoder.image.cvae, 21
- gnes.encoder.image.cvae\_cores, 20
- gnes.encoder.image.inception, 21
- gnes.encoder.image.inception\_cores, 20
- gnes.encoder.image.onnx, 22
- gnes.encoder.numeric, 23
- gnes.encoder.numeric.hash, 22
- gnes.encoder.numeric.pca, 22
- gnes.encoder.numeric.pq, 23
- gnes.encoder.numeric.tf\_pq, 23
- gnes.encoder.numeric.vlad, 23
- gnes.encoder.text, 26
- gnes.encoder.text.bert, 24
- gnes.encoder.text.elmo, 24
- gnes.encoder.text.flair, 25
- gnes.encoder.text.gpt, 25
- gnes.encoder.text.torch\_transformers, 25
- gnes.encoder.text.w2v, 26
- gnes.encoder.video, 27
- gnes.encoder.video.incep\_mixture, 27
- gnes.encoder.video.mixture\_core, 26
- gnes.helper, 42
- gnes.indexer, 32
- gnes.indexer.base, 31
- gnes.indexer.fulltext, 29
- gnes.indexer.fulltext.leveldb, 29
- gnes.indexer.key\_only, 32
- gnes.indexer.vector, 31
- gnes.indexer.vector.annoy, 30
- gnes.indexer.vector.faiss, 30
- gnes.indexer.vector.numpy, 30
- gnes.preprocessor, 36
- gnes.preprocessor.audio, 33
- gnes.preprocessor.audio.base, 33
- gnes.preprocessor.base, 35
- gnes.preprocessor.image, 34
- gnes.preprocessor.image.base, 33
- gnes.preprocessor.image.resize, 33
- gnes.preprocessor.image.segmentation, 34
- gnes.preprocessor.text, 34
- gnes.preprocessor.text.base, 34
- gnes.preprocessor.text.simple, 34
- gnes.preprocessor.video, 35
- gnes.preprocessor.video.base, 35
- gnes.proto, 36
- gnes.proto.gnes\_pb2, 36
- gnes.proto.gnes\_pb2\_grpc, 36
- gnes.router, 40
- gnes.router.base, 39
- gnes.router.map, 37
- gnes.router.map.simple, 37
- gnes.router.reduce, 39
- gnes.router.reduce.chunk, 37
- gnes.router.reduce.chunk\_sum, 38
- gnes.router.reduce.concat, 38
- gnes.router.reduce.doc\_sum, 38
- gnes.router.reduce.document, 38

`gnes.service`, 42  
`gnes.service.base`, 40  
`gnes.service.encoder`, 41  
`gnes.service.frontend`, 42  
`gnes.service.grpc`, 42  
`gnes.service.indexer`, 42  
`gnes.service.preprocessor`, 42  
`gnes.service.router`, 42

## A

- `acquire()` (*gnes.helper.FileLock* method), 43
- `add()` (*gnes.indexer.base.BaseIndexer* method), 31
- `add()` (*gnes.indexer.base.BaseKeyIndexer* method), 31
- `add()` (*gnes.indexer.base.BaseTextIndexer* method), 31
- `add()` (*gnes.indexer.base.BaseVectorIndexer* method), 31
- `add()` (*gnes.indexer.base.JointIndexer* method), 31
- `add()` (*gnes.indexer.fulltext.leveldb.AsyncLVDBIndexer* method), 29
- `add()` (*gnes.indexer.fulltext.leveldb.LVDBIndexer* method), 29
- `add()` (*gnes.indexer.key\_only.DictKeyIndexer* method), 32
- `add()` (*gnes.indexer.key\_only.ListKeyIndexer* method), 32
- `add()` (*gnes.indexer.key\_only.ListNumpyKeyIndexer* method), 32
- `add()` (*gnes.indexer.key\_only.NumpyKeyIndexer* method), 32
- `add()` (*gnes.indexer.vector.annoy.AnnoyIndexer* method), 30
- `add()` (*gnes.indexer.vector.faiss.FaissIndexer* method), 30
- `add()` (*gnes.indexer.vector.numpy.NumpyIndexer* method), 30
- `add_comp()` (*gnes.composer.base.YamlComposer* method), 19
- `add_GnesRPCServicer_to_server()` (in module *gnes.proto.gnes\_pb2\_grpc*), 36
- `add_layer()` (*gnes.composer.base.YamlComposer* method), 19
- `add_modules()` (*gnes.helper.PathImporter* static method), 43
- `add_route()` (in module *gnes.proto*), 36
- `ALWAYS_ONE` (*gnes.service.base.ReduceOp* attribute), 41
- `AnnoyIndexer` (class in *gnes.indexer.vector.annoy*), 30
- `append()` (*gnes.composer.base.YamlComposer.Layer* method), 19
- `apply()` (*gnes.preprocessor.base.BasePreprocessor* method), 35
- `apply()` (*gnes.preprocessor.base.PipelinePreprocessor* method), 35
- `apply()` (*gnes.preprocessor.base.UnaryPreprocessor* method), 35
- `apply()` (*gnes.preprocessor.image.resize.ResizeChunkPreprocessor* method), 33
- `apply()` (*gnes.preprocessor.image.segmentation.SegmentPreprocessor* method), 34
- `apply()` (*gnes.preprocessor.text.simple.TextPreprocessor* method), 34
- `apply()` (*gnes.router.base.BaseMapRouter* method), 39
- `apply()` (*gnes.router.base.BaseReduceRouter* method), 39
- `apply()` (*gnes.router.base.BaseRouter* method), 39
- `apply()` (*gnes.router.map.simple.DocBatchRouter* method), 37
- `apply()` (*gnes.router.map.simple.PublishRouter* method), 37
- `apply()` (*gnes.router.reduce.chunk.ChunkToDocumentRouter* method), 37
- `apply()` (*gnes.router.reduce.chunk\_sum.ChunkSumRouter* method), 38
- `apply()` (*gnes.router.reduce.concat.ConcatEmbedRouter* method), 38
- `apply()` (*gnes.router.reduce.doc\_sum.DocSumRouter* method), 38
- `apply()` (*gnes.router.reduce.document.DocFillRouter* method), 38
- `array2blob()` (in module *gnes.proto*), 36
- `AsyncLVDBIndexer` (class in *gnes.indexer.fulltext.leveldb*), 29

## B

- `BaseAudioEncoder` (class in *gnes.encoder.base*), 27
- `BaseAudioPreprocessor` (class in *gnes.preprocessor.audio.base*), 33
- `BaseBinaryEncoder` (class in *gnes.encoder.base*), 27

- BaseEncoder (class in *gnes.encoder.base*), 27
- BaseImageEncoder (class in *gnes.encoder.base*), 28
- BaseImagePreprocessor (class in *gnes.preprocessor.image.base*), 33
- BaseIndexer (class in *gnes.indexer.base*), 31
- BaseKeyIndexer (class in *gnes.indexer.base*), 31
- BaseMapRouter (class in *gnes.router.base*), 39
- BaseNumericEncoder (class in *gnes.encoder.base*), 28
- BaseONNXImageEncoder (class in *gnes.encoder.image.onnx*), 22
- BasePreprocessor (class in *gnes.preprocessor.base*), 35
- BasePytorchEncoder (class in *gnes.encoder.image.base*), 21
- BaseReduceRouter (class in *gnes.router.base*), 39
- BaseRouter (class in *gnes.router.base*), 39
- BaseService (class in *gnes.service.base*), 40
- BaseTextEncoder (class in *gnes.encoder.base*), 28
- BaseTextIndexer (class in *gnes.indexer.base*), 31
- BaseTextPreprocessor (class in *gnes.preprocessor.text.base*), 34
- BaseVectorIndexer (class in *gnes.indexer.base*), 31
- BaseVideoEncoder (class in *gnes.encoder.base*), 28
- BaseVideoPreprocessor (class in *gnes.preprocessor.video.base*), 35
- batch\_iterator() (in module *gnes.helper*), 43
- batch\_size (*gnes.encoder.audio.mfcc.MfccEncoder* attribute), 20
- batch\_size (*gnes.encoder.image.base.BasePytorchEncoder* attribute), 21
- batch\_size (*gnes.encoder.image.cvae.CVAEEncoder* attribute), 21
- batch\_size (*gnes.encoder.image.inception.TFInceptionEncoder* attribute), 21
- batch\_size (*gnes.encoder.image.onnx.BaseONNXImageEncoder* attribute), 22
- batch\_size (*gnes.encoder.numeric.hash.HashEncoder* attribute), 22
- batch\_size (*gnes.encoder.numeric.pca.PCALocalEncoder* attribute), 22
- batch\_size (*gnes.encoder.numeric.pq.PQEncoder* attribute), 23
- batch\_size (*gnes.encoder.numeric.tf\_pq.TFPQEncoder* attribute), 23
- batch\_size (*gnes.encoder.numeric.vlad.VladEncoder* attribute), 23
- batch\_size (*gnes.encoder.text.elmo.ElmoEncoder* attribute), 24
- batch\_size (*gnes.encoder.text.gpt.GPTEncoder* attribute), 25
- batch\_size (*gnes.encoder.video.incep\_mixture.IncepMixtureEncoder* attribute), 27
- batching() (in module *gnes.helper*), 43
- BenchmarkClient (class in *gnes.client.benchmark*), 18
- BertEncoder (class in *gnes.encoder.text.bert*), 24
- BertEncoderServer (class in *gnes.encoder.text.bert*), 24
- BertEncoderWithServer (class in *gnes.encoder.text.bert*), 24
- BetterEnum (class in *gnes.service.base*), 40
- blob2array() (in module *gnes.proto*), 36
- BlockMessage, 40
- build\_all() (*gnes.composer.base.YamlComposer* method), 19
- build\_dockerswarm() (*gnes.composer.base.YamlComposer* static method), 19
- build\_html() (*gnes.composer.base.YamlComposer* static method), 19
- build\_kubernetes() (*gnes.composer.base.YamlComposer* static method), 19
- build\_layers() (*gnes.composer.base.YamlComposer* method), 19
- build\_mermaid() (*gnes.composer.base.YamlComposer* static method), 19
- build\_shell() (*gnes.composer.base.YamlComposer* static method), 19
- build\_socket() (in module *gnes.service.base*), 41
- ## C
- call() (*gnes.proto.gnes\_pb2\_grpc.GnesRPCService* method), 36
- capacity (*gnes.indexer.key\_only.NumpyKeyIndexer* attribute), 32
- check\_fields() (*gnes.composer.base.YamlComposer* method), 19
- ChunkSumRouter (class in *gnes.router.reduce.chunk\_sum*), 38
- ChunkToDocumentRouter (class in *gnes.router.reduce.chunk*), 37
- CLIClient (class in *gnes.client.cli*), 18
- client() (in module *gnes.cli.api*), 17
- close() (*gnes.base.CompositionalTrainableBase* method), 17
- close() (*gnes.base.TrainableBase* method), 16
- close() (*gnes.client.base.ZmqClient* method), 18
- close() (*gnes.encoder.numeric.tf\_pq.TFPQEncoder* method), 23
- close() (*gnes.encoder.text.bert.BertEncoder* method), 24
- close() (*gnes.encoder.text.bert.BertEncoderServer* method), 24
- close() (*gnes.indexer.fulltext.leveldb.AsyncLVDBIndexer* method), 29

- close () (*gnes.indexer.fulltext.leveldb.LVDBIndexer method*), 29
- close () (*gnes.service.base.BaseService method*), 40
- close () (*gnes.service.grpc.GRPCService method*), 42
- comp2args (*gnes.composer.base.YamlComposer attribute*), 19
- comp2file (*gnes.composer.base.YamlComposer attribute*), 19
- ComponentNotLoad, 40
- components (*gnes.base.CompositionalTrainableBase attribute*), 17
- components (*gnes.indexer.base.JointIndexer attribute*), 31
- compose () (*in module gnes.cli.api*), 17
- CompositionalTrainableBase (*class in gnes.base*), 17
- CONCAT (*gnes.service.base.ReduceOp attribute*), 41
- ConcatEmbedRouter (*class in gnes.router.reduce.concat*), 38
- ConcurrentService (*class in gnes.service.base*), 40
- CVAEEncoder (*class in gnes.encoder.image.cvae*), 21
- D**
- default\_host (*gnes.service.base.BaseService attribute*), 40
- default\_values (*gnes.composer.base.YamlComposer.Layer attribute*), 19
- DictKeyIndexer (*class in gnes.indexer.key\_only*), 32
- doc\_type (*gnes.preprocessor.audio.base.BaseAudioPreprocessor attribute*), 33
- doc\_type (*gnes.preprocessor.base.BasePreprocessor attribute*), 35
- doc\_type (*gnes.preprocessor.image.base.BaseImagePreprocessor attribute*), 33
- doc\_type (*gnes.preprocessor.text.base.BaseTextPreprocessor attribute*), 34
- doc\_type (*gnes.preprocessor.video.base.BaseVideoPreprocessor attribute*), 35
- DocBatchRouter (*class in gnes.router.map.simple*), 37
- DocFillRouter (*class in gnes.router.reduce.document*), 38
- DocSumRouter (*class in gnes.router.reduce.doc\_sum*), 38
- dump () (*gnes.base.TrainableBase method*), 16
- dump () (*gnes.service.base.BaseService method*), 40
- dump\_full\_path (*gnes.base.TrainableBase attribute*), 16
- dump\_yaml () (*gnes.base.TrainableBase method*), 16
- E**
- ElmoEncoder (*class in gnes.encoder.text.elmo*), 24
- encode () (*gnes.encoder.audio.mfcc.MfccEncoder method*), 20
- encode () (*gnes.encoder.base.BaseAudioEncoder method*), 27
- encode () (*gnes.encoder.base.BaseBinaryEncoder method*), 27
- encode () (*gnes.encoder.base.BaseEncoder method*), 27
- encode () (*gnes.encoder.base.BaseImageEncoder method*), 28
- encode () (*gnes.encoder.base.BaseNumericEncoder method*), 28
- encode () (*gnes.encoder.base.BaseTextEncoder method*), 28
- encode () (*gnes.encoder.base.BaseVideoEncoder method*), 28
- encode () (*gnes.encoder.base.PipelineEncoder method*), 28
- encode () (*gnes.encoder.image.base.BasePytorchEncoder method*), 21
- encode () (*gnes.encoder.image.cvae.CVAEEncoder method*), 21
- encode () (*gnes.encoder.image.inception.TFInceptionEncoder method*), 21
- encode () (*gnes.encoder.image.onnx.BaseONNXImageEncoder method*), 22
- encode () (*gnes.encoder.numeric.hash.HashEncoder method*), 22
- encode () (*gnes.encoder.numeric.pca.PCALocalEncoder method*), 22
- encode () (*gnes.encoder.numeric.pq.PQEncoder method*), 23
- encode () (*gnes.encoder.numeric.tf\_pq.TFPQEncoder method*), 23
- encode () (*gnes.encoder.numeric.vlad.VladEncoder method*), 23
- encode () (*gnes.encoder.text.bert.BertEncoder method*), 24
- encode () (*gnes.encoder.text.bert.BertEncoderWithServer method*), 24
- encode () (*gnes.encoder.text.elmo.ElmoEncoder method*), 24
- encode () (*gnes.encoder.text.flair.FlairEncoder method*), 25
- encode () (*gnes.encoder.text.gpt.GPTEncoder method*), 25
- encode () (*gnes.encoder.text.torch\_transformers.TorchTransformersEncoder method*), 25
- encode () (*gnes.encoder.text.w2v.Word2VecEncoder method*), 26
- encode () (*gnes.encoder.video.incep\_mixture.IncepMixtureEncoder method*), 27
- encode () (*in module gnes.cli.api*), 17
- EncoderService (*class in gnes.service.encoder*), 41
- EventLoopEnd, 40

## F

FaissIndexer (class in *gnes.indexer.vector.faiss*), 30  
 FileLock (class in *gnes.helper*), 43  
 FlairEncoder (class in *gnes.encoder.text.flair*), 25  
 from\_string (*gnes.service.base.BetterEnum* attribute), 40  
 from\_yaml () (*gnes.base.CompositionalTrainableBase* class method), 17  
 from\_yaml () (*gnes.base.TrainableBase* class method), 16  
 frontend () (in module *gnes.cli.api*), 17  
 FrontendService (class in *gnes.service.frontend*), 42

## G

get\_chunks\_from\_docs () (*gnes.service.encoder.EncoderService* static method), 41  
 get\_component\_name (*gnes.composer.base.YamlComposer.Layer* attribute), 19  
 get\_first\_available\_gpu () (in module *gnes.helper*), 43  
 get\_main\_parser () (in module *gnes.cli.parser*), 17  
 get\_optimal\_sample\_size () (in module *gnes.helper*), 42  
 get\_perm () (in module *gnes.helper*), 43  
 get\_sys\_info () (in module *gnes.helper*), 42  
 get\_value () (*gnes.composer.base.YamlComposer.Layer* static method), 19  
 gnes (module), 43  
 gnes.base (module), 16  
 gnes.cli (module), 18  
 gnes.cli.api (module), 17  
 gnes.cli.parser (module), 17  
 gnes.client (module), 19  
 gnes.client.base (module), 18  
 gnes.client.benchmark (module), 18  
 gnes.client.cli (module), 18  
 gnes.client.http (module), 18  
 gnes.composer (module), 20  
 gnes.composer.base (module), 19  
 gnes.composer.flask (module), 19  
 gnes.encoder (module), 28  
 gnes.encoder.audio (module), 20  
 gnes.encoder.audio.mfcc (module), 20  
 gnes.encoder.base (module), 27  
 gnes.encoder.image (module), 22  
 gnes.encoder.image.base (module), 21  
 gnes.encoder.image.cvae (module), 21  
 gnes.encoder.image.cvae\_cores (module), 20  
 gnes.encoder.image.inception (module), 21  
 gnes.encoder.image.inception\_cores (module), 20  
 gnes.encoder.image.onnx (module), 22

*gnes.encoder.numeric* (module), 23  
*gnes.encoder.numeric.hash* (module), 22  
*gnes.encoder.numeric.pca* (module), 22  
*gnes.encoder.numeric.pq* (module), 23  
*gnes.encoder.numeric.tf\_pq* (module), 23  
*gnes.encoder.numeric.vlad* (module), 23  
*gnes.encoder.text* (module), 26  
*gnes.encoder.text.bert* (module), 24  
*gnes.encoder.text.elmo* (module), 24  
*gnes.encoder.text.flair* (module), 25  
*gnes.encoder.text.gpt* (module), 25  
*gnes.encoder.text.torch\_transformers* (module), 25  
*gnes.encoder.text.w2v* (module), 26  
*gnes.encoder.video* (module), 27  
*gnes.encoder.video.incep\_mixture* (module), 27  
*gnes.encoder.video.mixture\_core* (module), 26  
*gnes.helper* (module), 42  
*gnes.indexer* (module), 32  
*gnes.indexer.base* (module), 31  
*gnes.indexer.fulltext* (module), 29  
*gnes.indexer.fulltext.leveldb* (module), 29  
*gnes.indexer.key\_only* (module), 32  
*gnes.indexer.vector* (module), 31  
*gnes.indexer.vector.annoy* (module), 30  
*gnes.indexer.vector.faiss* (module), 30  
*gnes.indexer.vector.numpy* (module), 30  
*gnes.preprocessor* (module), 36  
*gnes.preprocessor.audio* (module), 33  
*gnes.preprocessor.audio.base* (module), 33  
*gnes.preprocessor.base* (module), 35  
*gnes.preprocessor.image* (module), 34  
*gnes.preprocessor.image.base* (module), 33  
*gnes.preprocessor.image.resize* (module), 33  
*gnes.preprocessor.image.segmentation* (module), 34  
*gnes.preprocessor.text* (module), 34  
*gnes.preprocessor.text.base* (module), 34  
*gnes.preprocessor.text.simple* (module), 34  
*gnes.preprocessor.video* (module), 35  
*gnes.preprocessor.video.base* (module), 35  
*gnes.proto* (module), 36  
*gnes.proto.gnes\_pb2* (module), 36  
*gnes.proto.gnes\_pb2\_grpc* (module), 36  
*gnes.router* (module), 40  
*gnes.router.base* (module), 39  
*gnes.router.map* (module), 37  
*gnes.router.map.simple* (module), 37  
*gnes.router.reduce* (module), 39  
*gnes.router.reduce.chunk* (module), 37  
*gnes.router.reduce.chunk\_sum* (module), 38



gnes.router.reduce.concat (*module*), 38  
 gnes.router.reduce.doc\_sum (*module*), 38  
 gnes.router.reduce.document (*module*), 38  
 gnes.service (*module*), 42  
 gnes.service.base (*module*), 40  
 gnes.service.encoder (*module*), 41  
 gnes.service.frontend (*module*), 42  
 gnes.service.grpc (*module*), 42  
 gnes.service.indexer (*module*), 42  
 gnes.service.preprocessor (*module*), 42  
 gnes.service.router (*module*), 42  
 GnesRPCServicer (class in *gnes.proto.gnes\_pb2\_grpc*), 36  
 GnesRPCStub (class in *gnes.proto.gnes\_pb2\_grpc*), 36  
 GPT2Encoder (class in *gnes.encoder.text.gpt*), 25  
 GPTEncoder (class in *gnes.encoder.text.gpt*), 25  
 grpc () (in *module gnes.cli.api*), 17  
 GRPCService (class in *gnes.service.grpc*), 42

## H

handler (*gnes.service.base.BaseService attribute*), 40  
 handler (*gnes.service.encoder.EncoderService attribute*), 41  
 handler (*gnes.service.grpc.GRPCService attribute*), 42  
 handler (*gnes.service.indexer.IndexerService attribute*), 42  
 handler (*gnes.service.preprocessor.PreprocessorService attribute*), 42  
 handler (*gnes.service.router.RouterService attribute*), 42  
 hash () (*gnes.encoder.numeric.hash.HashEncoder method*), 22  
 HashEncoder (class in *gnes.encoder.numeric.hash*), 22  
 HttpClient (class in *gnes.client.http*), 18

## I

IncepMixtureEncoder (class in *gnes.encoder.video.incep\_mixture*), 27  
 Index () (*gnes.proto.gnes\_pb2\_grpc.GnesRPCServicer method*), 36  
 index () (*gnes.proto.RequestGenerator static method*), 36  
 index () (in *module gnes.cli.api*), 17  
 IndexerService (class in *gnes.service.indexer*), 42  
 is\_bind (*gnes.service.base.SocketType attribute*), 41  
 is\_block (*gnes.service.base.ParallelType attribute*), 41  
 is\_heto\_single\_component (*gnes.composer.base.YamlComposer.Layer attribute*), 19  
 is\_homo\_multi\_component (*gnes.composer.base.YamlComposer.Layer attribute*), 19

is\_homogenous (*gnes.composer.base.YamlComposer.Layer attribute*), 19  
 is\_locked (*gnes.helper.FileLock attribute*), 43  
 is\_pipeline (*gnes.base.CompositionalTrainableBase attribute*), 17  
 is\_push (*gnes.service.base.ParallelType attribute*), 41  
 is\_single\_component (*gnes.composer.base.YamlComposer.Layer attribute*), 19  
 is\_trained (*gnes.encoder.text.bert.BertEncoder attribute*), 24  
 is\_trained (*gnes.encoder.text.bert.BertEncoderServer attribute*), 24  
 is\_trained (*gnes.encoder.text.elmo.ElmoEncoder attribute*), 24  
 is\_trained (*gnes.encoder.text.flair.FlairEncoder attribute*), 25  
 is\_trained (*gnes.encoder.text.gpt.GPTEncoder attribute*), 25  
 is\_trained (*gnes.encoder.text.torch\_transformers.TorchTransformersEncoder attribute*), 26  
 is\_trained (*gnes.encoder.text.w2v.Word2VecEncoder attribute*), 26  
 is\_trained (*gnes.preprocessor.base.UnaryPreprocessor attribute*), 35

## J

join () (*gnes.service.base.ServiceManager method*), 41  
 JointIndexer (class in *gnes.indexer.base*), 31

## K

kmeans\_pred () (*gnes.encoder.numeric.vlad.VladEncoder method*), 23  
 kmeans\_train () (*gnes.encoder.numeric.vlad.VladEncoder method*), 23

## L

ListKeyIndexer (class in *gnes.indexer.key\_only*), 32  
 ListNumpyKeyIndexer (class in *gnes.indexer.key\_only*), 32  
 load () (*gnes.base.TrainableBase static method*), 16  
 load\_contrib\_module () (in *module gnes.helper*), 43  
 load\_model () (*gnes.service.base.BaseService method*), 40  
 load\_yaml () (*gnes.base.TrainableBase class method*), 16  
 LVDBIndexer (class in *gnes.indexer.fulltext.leveldb*), 29

## M

main () (in *module gnes.cli*), 18  
 message\_handler () (*gnes.service.base.BaseService method*), 40

MessageHandler (class in *gnes.service.base*), 40  
 MfccEncoder (class in *gnes.encoder.audio.mfcc*), 20

## N

normalize\_score() (*gnes.indexer.base.BaseIndexer* method), 31  
 normalize\_score() (*gnes.indexer.vector.annoy.annoyIndexer* method), 30  
 normalize\_score() (*gnes.indexer.vector.faiss.FaissIndexer* method), 30  
 NumpyIndexer (class in *gnes.indexer.vector.numpy*), 30  
 NumpyKeyIndexer (class in *gnes.indexer.key\_only*), 32

## P

PAIR\_BIND (*gnes.service.base.SocketType* attribute), 41  
 PAIR\_CONNECT (*gnes.service.base.SocketType* attribute), 41  
 ParallelType (class in *gnes.service.base*), 40  
 parse\_arg() (in module *gnes.helper*), 43  
 PathImporter (class in *gnes.helper*), 43  
 PCALocalEncoder (class in *gnes.encoder.numeric.pca*), 22  
 PipelineEncoder (class in *gnes.encoder.base*), 28  
 PipelinePreprocessor (class in *gnes.preprocessor.base*), 35  
 post\_init() (*gnes.base.TrainableBase* method), 16  
 post\_init() (*gnes.encoder.image.base.BasePytorchEncoder* method), 21  
 post\_init() (*gnes.encoder.image.cvae.CVAEEncoder* method), 21  
 post\_init() (*gnes.encoder.image.inception.TFInceptionEncoder* method), 21  
 post\_init() (*gnes.encoder.image.onnx.BaseONNXImageEncoder* method), 22  
 post\_init() (*gnes.encoder.numeric.tf\_pq.TFPQEncoder* method), 23  
 post\_init() (*gnes.encoder.text.bert.BertEncoder* method), 24  
 post\_init() (*gnes.encoder.text.bert.BertEncoderServer* method), 24  
 post\_init() (*gnes.encoder.text.elmo.ElmoEncoder* method), 24  
 post\_init() (*gnes.encoder.text.flair.FlairEncoder* method), 25  
 post\_init() (*gnes.encoder.text.gpt.GPTEncoder* method), 25  
 post\_init() (*gnes.encoder.text.torch\_transformers.TorchTransformersEncoder* method), 26

post\_init() (*gnes.encoder.text.w2v.Word2VecEncoder* method), 26  
 post\_init() (*gnes.encoder.video.incep\_mixture.IncepMixtureEncoder* method), 27  
 post\_init() (*gnes.indexer.fulltext.leveldb.AsyncLVDBIndexer* method), 29  
 post\_init() (*gnes.indexer.fulltext.leveldb.LVDBIndexer* method), 29  
 post\_init() (*gnes.indexer.vector.annoy.annoyIndexer* method), 30  
 post\_init() (*gnes.indexer.vector.faiss.FaissIndexer* method), 30  
 post\_init() (*gnes.preprocessor.image.segmentation.SegmentPreprocessor* method), 34  
 post\_init() (*gnes.service.base.BaseService* method), 40  
 post\_init() (*gnes.service.encoder.EncoderService* method), 41  
 post\_init() (*gnes.service.grpc.GRPCService* method), 42  
 post\_init() (*gnes.service.indexer.IndexerService* method), 42  
 post\_init() (*gnes.service.preprocessor.PreprocessorService* method), 42  
 post\_init() (*gnes.service.router.RouterService* method), 42  
 PQEncoder (class in *gnes.encoder.numeric.pq*), 23  
 pre\_init() (*gnes.base.TrainableBase* class method), 16  
 pre\_init() (*gnes.encoder.numeric.tf\_pq.TFPQEncoder* class method), 23  
 pred\_kmeans() (*gnes.encoder.numeric.hash.HashEncoder* method), 22  
 preprocess() (in module *gnes.cli.api*), 17  
 PreprocessorService (class in *gnes.service.preprocessor*), 42  
 profiling() (in module *gnes.helper*), 43  
 PUB\_BIND (*gnes.service.base.SocketType* attribute), 41  
 PUB\_BLOCK (*gnes.service.base.ParallelType* attribute), 40  
 PUB\_CONNECT (*gnes.service.base.SocketType* attribute), 41  
 PUB\_NONBLOCK (*gnes.service.base.ParallelType* attribute), 40  
 PublishRouter (class in *gnes.router.map.simple*), 37  
 PULL\_BIND (*gnes.service.base.SocketType* attribute), 41  
 PULL\_CONNECT (*gnes.service.base.SocketType* attribute), 41  
 PUSH\_BIND (*gnes.service.base.SocketType* attribute), 41  
 PUSH\_BLOCK (*gnes.service.base.ParallelType* attribute), 40  
 PUSH\_CONNECT (*gnes.service.base.SocketType* at-

tribute), 41  
 PUSH\_NONBLOCK (*gnes.service.base.ParallelType* attribute), 41

## Q

query() (*gnes.indexer.base.BaseIndexer* method), 31  
 query() (*gnes.indexer.base.BaseKeyIndexer* method), 31  
 query() (*gnes.indexer.base.BaseTextIndexer* method), 31  
 query() (*gnes.indexer.base.BaseVectorIndexer* method), 31  
 query() (*gnes.indexer.base.JointIndexer* method), 31  
 query() (*gnes.indexer.fulltext.leveldb.AsyncLVDBIndexer* method), 29  
 query() (*gnes.indexer.fulltext.leveldb.LVDBIndexer* method), 29  
 query() (*gnes.indexer.key\_only.DictKeyIndexer* method), 32  
 query() (*gnes.indexer.key\_only.ListKeyIndexer* method), 32  
 query() (*gnes.indexer.key\_only.ListNumpyKeyIndexer* method), 32  
 query() (*gnes.indexer.key\_only.NumpyKeyIndexer* method), 32  
 query() (*gnes.indexer.vector.annoy.AnnoyIndexer* method), 30  
 query() (*gnes.indexer.vector.faiss.FaissIndexer* method), 30  
 query() (*gnes.indexer.vector.numpy.NumpyIndexer* method), 30  
 Query() (*gnes.proto.gnes\_pb2\_grpc.GnesRPCService* method), 36  
 query() (*gnes.proto.RequestGenerator* static method), 36

## R

ran\_gen() (*gnes.encoder.numeric.hash.HashEncoder* method), 22  
 raw\_to\_chunk() (*gnes.preprocessor.base.UnaryPreprocessor* method), 35  
 recv\_message() (*gnes.client.base.ZmqClient* method), 18  
 recv\_message() (*in module gnes.proto*), 36  
 ReduceOp (*class in gnes.service.base*), 41  
 register() (*gnes.service.base.MessageHandler* method), 40  
 release() (*gnes.helper.FileLock* method), 43  
 RequestGenerator (*class in gnes.proto*), 36  
 ResizeChunkPreprocessor (*class in gnes.preprocessor.image.resize*), 33  
 resolve\_py\_path() (*in module gnes.cli.parser*), 17  
 resolve\_yaml\_path() (*in module gnes.cli.parser*), 17

route() (*in module gnes.cli.api*), 17  
 RouterService (*class in gnes.service.router*), 42  
 run() (*gnes.composer.flask.YamlComposerFlask* method), 19  
 run() (*gnes.service.base.BaseService* method), 40

## S

SegmentPreprocessor (*class in gnes.preprocessor.image.segmentation*), 34  
 send\_ctrl\_message() (*in module gnes.service.base*), 41  
 send\_message() (*gnes.client.base.ZmqClient* method), 18  
 send\_message() (*in module gnes.proto*), 36  
 serve() (*gnes.service.base.MessageHandler* method), 40  
 ServiceError, 41  
 ServiceManager (*class in gnes.service.base*), 41  
 set\_base\_parser() (*in module gnes.cli.parser*), 17  
 set\_client\_benchmark\_parser() (*in module gnes.cli.parser*), 17  
 set\_client\_cli\_parser() (*in module gnes.cli.parser*), 17  
 set\_client\_http\_parser() (*in module gnes.cli.parser*), 17  
 set\_composer\_flask\_parser() (*in module gnes.cli.parser*), 17  
 set\_composer\_parser() (*in module gnes.cli.parser*), 17  
 set\_encoder\_parser() (*in module gnes.cli.parser*), 18  
 set\_frontend\_parser() (*in module gnes.cli.parser*), 18  
 set\_grpc\_service\_parser() (*in module gnes.cli.parser*), 18  
 set\_indexer\_parser() (*in module gnes.cli.parser*), 18  
 set\_logger() (*in module gnes.helper*), 43  
 set\_preprocessor\_parser() (*in module gnes.cli.parser*), 18  
 set\_router\_parser() (*in module gnes.cli.parser*), 18  
 set\_service\_parser() (*in module gnes.cli.parser*), 18  
 size (*gnes.indexer.key\_only.DictKeyIndexer* attribute), 32  
 size (*gnes.indexer.key\_only.ListKeyIndexer* attribute), 32  
 size (*gnes.indexer.key\_only.NumpyKeyIndexer* attribute), 32  
 size (*gnes.indexer.vector.annoy.AnnoyIndexer* attribute), 30  
 size (*gnes.indexer.vector.faiss.FaissIndexer* attribute), 30

SocketType (class in *gnes.service.base*), 41  
 start() (*gnes.client.http.HttpClient* method), 18  
 status (*gnes.service.base.BaseService* attribute), 40  
 store\_args\_kwargs (*gnes.base.TrainableBase* attribute), 16  
 store\_args\_kwargs  
     (*gnes.encoder.text.bert.BertEncoder* attribute), 24  
 store\_args\_kwargs  
     (*gnes.encoder.text.bert.BertEncoderServer* attribute), 24  
 StreamCall() (*gnes.proto.gnes\_pb2\_grpc.GnesRPCServicer* method), 36  
 SUB\_BIND (*gnes.service.base.SocketType* attribute), 41  
 SUB\_CONNECT (*gnes.service.base.SocketType* attribute), 41

## T

TextPreprocessor (class in *gnes.preprocessor.text.simple*), 34  
 TFInceptionEncoder (class in *gnes.encoder.image.inception*), 21  
 TFPQEncoder (class in *gnes.encoder.numeric.tf\_pq*), 23  
 time\_profile() (in module *gnes.helper*), 43  
 to\_yaml() (*gnes.base.CompositionalTrainableBase* class method), 17  
 to\_yaml() (*gnes.base.TrainableBase* class method), 16  
 TorchTransformersEncoder (class in *gnes.encoder.text.torch\_transformers*), 25  
 train() (*gnes.base.CompositionalTrainableBase* method), 17  
 train() (*gnes.base.TrainableBase* method), 16  
 train() (*gnes.encoder.audio.mfcc.MfccEncoder* method), 20  
 train() (*gnes.encoder.base.BaseAudioEncoder* method), 27  
 train() (*gnes.encoder.base.BaseBinaryEncoder* method), 27  
 train() (*gnes.encoder.base.BaseEncoder* method), 27  
 train() (*gnes.encoder.base.BaseImageEncoder* method), 28  
 train() (*gnes.encoder.base.BaseNumericEncoder* method), 28  
 train() (*gnes.encoder.base.BaseTextEncoder* method), 28  
 train() (*gnes.encoder.base.BaseVideoEncoder* method), 28  
 train() (*gnes.encoder.base.PipelineEncoder* method), 28  
 train() (*gnes.encoder.image.base.BasePytorchEncoder* method), 21  
 train() (*gnes.encoder.image.cvae.CVAEEncoder* method), 21  
 train() (*gnes.encoder.image.inception.TFInceptionEncoder* method), 21  
 train() (*gnes.encoder.image.onnx.BaseONNXImageEncoder* method), 22  
 train() (*gnes.encoder.numeric.hash.HashEncoder* method), 22  
 train() (*gnes.encoder.numeric.pca.PCALocalEncoder* method), 22  
 train() (*gnes.encoder.numeric.pq.PQEncoder* method), 23  
 train() (*gnes.encoder.numeric.tf\_pq.TFPQEncoder* method), 23  
 train() (*gnes.encoder.numeric.vlad.VladEncoder* method), 23  
 train() (*gnes.encoder.text.bert.BertEncoder* method), 24  
 train() (*gnes.encoder.text.bert.BertEncoderServer* method), 24  
 train() (*gnes.encoder.text.bert.BertEncoderWithServer* method), 24  
 train() (*gnes.encoder.text.elmo.ElmoEncoder* method), 24  
 train() (*gnes.encoder.text.flair.FlairEncoder* method), 25  
 train() (*gnes.encoder.text.gpt.GPT2Encoder* method), 25  
 train() (*gnes.encoder.text.gpt.GPTEncoder* method), 25  
 train() (*gnes.encoder.text.torch\_transformers.TorchTransformersEncoder* method), 26  
 train() (*gnes.encoder.text.w2v.Word2VecEncoder* method), 26  
 train() (*gnes.encoder.video.incep\_mixture.IncepMixtureEncoder* method), 27  
 train() (*gnes.indexer.base.BaseIndexer* method), 31  
 train() (*gnes.indexer.base.BaseKeyIndexer* method), 31  
 train() (*gnes.indexer.base.BaseTextIndexer* method), 31  
 train() (*gnes.indexer.base.BaseVectorIndexer* method), 31  
 train() (*gnes.indexer.base.JointIndexer* method), 31  
 train() (*gnes.indexer.fulltext.leveldb.AsyncLVDBIndexer* method), 29  
 train() (*gnes.indexer.fulltext.leveldb.LVDBIndexer* method), 29  
 train() (*gnes.indexer.key\_only.DictKeyIndexer* method), 32  
 train() (*gnes.indexer.key\_only.ListKeyIndexer* method), 32  
 train() (*gnes.indexer.key\_only.ListNumpyKeyIndexer* method), 32

[train\(\)](#) (*gnes.indexer.key\_only.NumpyKeyIndexer* method), 32  
[train\(\)](#) (*gnes.indexer.vector.annoy.AnnoyIndexer* method), 30  
[train\(\)](#) (*gnes.indexer.vector.faiss.FaissIndexer* method), 30  
[train\(\)](#) (*gnes.indexer.vector.numpy.NumpyIndexer* method), 31  
[train\(\)](#) (*gnes.preprocessor.audio.base.BaseAudioPreprocessor* method), 33  
[train\(\)](#) (*gnes.preprocessor.base.BasePreprocessor* method), 35  
[train\(\)](#) (*gnes.preprocessor.base.PipelinePreprocessor* method), 35  
[train\(\)](#) (*gnes.preprocessor.base.UnaryPreprocessor* method), 35  
[train\(\)](#) (*gnes.preprocessor.image.base.BaseImagePreprocessor* method), 33  
[train\(\)](#) (*gnes.preprocessor.image.resize.ResizeChunkPreprocessor* method), 33  
[train\(\)](#) (*gnes.preprocessor.image.segmentation.SegmentPreprocessor* method), 34  
[train\(\)](#) (*gnes.preprocessor.text.base.BaseTextPreprocessor* method), 34  
[train\(\)](#) (*gnes.preprocessor.text.simple.TextPreprocessor* method), 34  
[train\(\)](#) (*gnes.preprocessor.video.base.BaseVideoPreprocessor* method), 35  
[Train\(\)](#) (*gnes.proto.gnes\_pb2\_grpc.GnesRPCServicer* method), 36  
[train\(\)](#) (*gnes.proto.RequestGenerator* static method), 36  
[train\(\)](#) (*gnes.router.base.BaseMapRouter* method), 39  
[train\(\)](#) (*gnes.router.base.BaseReduceRouter* method), 39  
[train\(\)](#) (*gnes.router.base.BaseRouter* method), 39  
[train\(\)](#) (*gnes.router.map.simple.DocBatchRouter* method), 37  
[train\(\)](#) (*gnes.router.map.simple.PublishRouter* method), 37  
[train\(\)](#) (*gnes.router.reduce.chunk.ChunkToDocumentRouter* method), 37  
[train\(\)](#) (*gnes.router.reduce.chunk\_sum.ChunkSumRouter* method), 38  
[train\(\)](#) (*gnes.router.reduce.concat.ConcatEmbedRouter* method), 38  
[train\(\)](#) (*gnes.router.reduce.doc\_sum.DocSumRouter* method), 38  
[train\(\)](#) (*gnes.router.reduce.document.DocFillRouter* method), 39  
[train\\_kmeans\(\)](#) (*gnes.encoder.numeric.hash.HashEncoder* method), 22  
[train\\_required\(\)](#) (in module *gnes.helper*), 43  
[TrainableBase](#) (class in *gnes.base*), 16

**U**

[UnaryPreprocessor](#) (class in *gnes.preprocessor.base*), 35

**V**

[VladEncoder](#) (class in *gnes.encoder.numeric.vlad*), 23

**W**

[Word2VecEncoder](#) (class in *gnes.encoder.text.w2v*), 26

**Y**

[yaml\\_full\\_path](#) (*gnes.base.TrainableBase* attribute), 17  
[YamlComposer](#) (class in *gnes.composer.base*), 19

[YamlComposer.Layer](#) (class in *gnes.composer.base*), 19

[YamlComposerFlask](#) (class in *gnes.composer.flask*), 19

**Z**

[ZmqClient](#) (class in *gnes.client.base*), 18