
logging-ldp Documentation

Release 0.0.1

Cedric Dumay

May 20, 2019

Contents

1	Get It Now	3
2	API focus	5
	Python Module Index	7

A python 3 logging bundle to send logs using GELF on the [OVH Logs Data Platform](#). The following example shows how to send log over TCP/TLS input.

```
import logging
from logging_ldp.formatters import LDPGELFFormatter
from logging_ldp.handlers import LDPGELFTCPHandler

logger = logging.getLogger("ldp")
logger.setLevel(logging.DEBUG)

handler = LDPGELFTCPHandler(hostname="gra1.logs.ovh.com")
handler.setFormatter(LDPGELFFormatter(token="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"))
logger.addHandler(handler)
logger.debug("hello !")
```


CHAPTER 1

Get It Now

First, install logging-ldap using `pip`:

```
pip install -U logging-ldap
```


The api is many only an implementation of `logging-gelf`

2.1 `logging_ldp.formatters` — Formatters

class `logging_ldp.formatters.LDPGELFFormatter`

A subclass of `logging_ldp.GELFFormatter` to format `LogRecord` into GELF.

__init__ (*token*, *schema*=<`logging_ldp.schemas.LDPSchema`>, *null_character*=`True`, *JSONEncoder*=`json.JSONEncoder`, *exclude_patterns*=`None`)

A GELF formatter to format a `logging.LogRecord` into GELF.

Parameters

- **token** (*str*) – The LDP token (aka. `X-OVH-TOKEN`).
- **schema** (`logging_ldp.schemas.LDPSchema`) – The marshmallow schema to use to format data.
- **null_character** (*bool*) – Append a ‘0’ at the end of the string. It depends on the input used.
- **JSONEncoder** (`json.JSONEncoder`) – A custom json encoder to use.
- **exclude_patterns** (*list* / `None`) – List of regexp used to exclude keys

format (*record*)

Format the specified record into json using the schema which MUST inherit from `logging_ldp.schemas.LDPSchema` to support LDP casting type (see: [The field naming convention](#)).

Parameters **record** (`logging.LogRecord`) – Contains all the information pertinent to the event being logged.

Returns A JSON dump of the record.

Return type `str`

2.2 logging_ldp.handlers — Handlers

class logging_ldp.handlers.LDPGELFTCPSocketHandler

The *LDPGELFTCPSocketHandler*, which inherit from logging_gelf.GELFTCPSocketHandler, sends logging output to a TCP/TLS network socket.

__init__ (*hostname*)

Initialize a TCP/TLS connection to the given *hostname*.

Parameters *hostname* (*str*) – Hostname/FQDN to connect to.

2.3 logging_ldp.schemas — Schemas

class logging_ldp.schemas.LDPSchema

Schema which allow to specify a mapping for logging.LogRecord. It based on logging_gelf.schemas.GelfSchema. All schema MUST inherit from this.

static **_forge_key** (*key*, *value*)

Allow to rename keys to cast types (see: [The field naming convention](#)).

Parameters

- **key** (*str*) – The attribute key
- **value** (*Any*) – The attribute value

Returns The key suffixed

Return type str

I

`logging_ldp.formatters`, 5
`logging_ldp.handlers`, 6
`logging_ldp.schemas`, 6

Symbols

`__init__()` (*logging_ldp.formatters.LDPGELFFormatter*
method), 5

`__init__()` (*logging_ldp.handlers.LDPGELFTCPSocketHandler*
method), 6

`_forge_key()` (*logging_ldp.schemas.LDPSchema*
static method), 6

F

`format()` (*logging_ldp.formatters.LDPGELFFormatter*
method), 5

L

`LDPGELFFormatter` (class in *log-*
ging_ldp.formatters), 5

`LDPGELFTCPSocketHandler` (class in *log-*
ging_ldp.handlers), 6

`LDPSchema` (class in *logging_ldp.schemas*), 6

`logging_ldp.formatters` (module), 5

`logging_ldp.handlers` (module), 6

`logging_ldp.schemas` (module), 6