TuxEatPi Common Documentation

Release 0.1

TuxEatPi Team

Sep 20, 2017

Contents:

1 tuxeatpi_common **1**
1.1 tuxeatpi_common package 1

2 Indices and tables **7**

Python Module Index **9**

CHAPTER 1

tuxeatpi_common

1.1 tuxeatpi_common package

1.1.1 Submodules

tuxeatpi_common.cli module

TuxEatPi Base Main module

`tuxeatpi_common.cli.`**`cli`**(*daemon_class=<class 'tuxeatpi_common.daemon.TepBaseDaemon'>*)

Main function to call the cli

`tuxeatpi_common.cli.`**`set_daemon_class`**(*class_object*)

Set DAEMON_CLASS global variable

This variable is used in CliTemplate fonction

tuxeatpi_common.daemon module

Module defining Base Daemon for TuxEatPi daemons

`class` `tuxeatpi_common.daemon.`**`TepBaseDaemon`**(*name*, *workdir*, *intents_folder*, *dialog_folder*, *logging_level=20*)

Bases: `object`

Base Daemon for TuxEatPi

`get_dialog`(*key*, ***kwargs*)

Get dialog and render it

`help_`()

Return help for this daemon

`main_loop`()

Main loop

Could be ReImplemented for advanced component

publish(*message*, *override_topic=None*, *qos=0*)
Publish message to MQTT

reload()
Reload the daemon

set_config(*config*)
Save the configuration and reload the daemon

Returns:

- True if the configuration looks good
- False otherwise

shutdown()
Shutdown the daemon form mqtt message

signal_handler(*signal_*, *frame*)
Signal handler

start()
Startup function for main loop

wait_for_speaking(*timeout=30*)
Wait for start and end speech

tuxeatpi_common.dialogs module

Module defining Dialog handler for voice

class tuxeatpi_common.dialogs.**DialogsHandler**(*dialog_folder*, *component_name*)
Bases: object

Class getting dialog for voice

get_dialog(*language*, *key*, ***kwargs*)
Return one sentente related to a key and a language

load()
Load dialogs from dialog folder

tuxeatpi_common.error module

Module defining TuxEatPi errors

exception tuxeatpi_common.error.**TuxEatPiError**
Bases: Exception

Base class for base exceptions

tuxeatpi_common.etcd_client module

Module defintion function to get etcd client

class tuxeatpi_common.etcd_client.**EtcdWrapper**(*host=None*, *port=None*)
Bases: object

Etcd Wrapper class to handle sync and async requests

async_read(*key*, *wait=False*)
Async Etcd read operation

delete(*key*, *recursive=False*)
Sync Etcd delete operation

eternal_watch(*key*, *recursive=False*)
Sync Etcd watch operation

read(*key*, *recursive=False*, *wait=False*, *timeout=60*)
Sync Etcd read operation

write(*key*, *value*, *serialize=True*)
Sync Etcd write operation

tuxeatpi_common.initializer module

Module defining the init process for TuxEatPi component

class tuxeatpi_common.initializer.**Initializer**(*component*, *skip_dialogs=False*, *skip_intents=False*, *skip_settings=False*)

Bases: object

Initializer class to run init action for a component

run()
Run initialization

tuxeatpi_common.intents module

Module defining how to handle intents

class tuxeatpi_common.intents.**IntentsHandler**(*intent_folder*, *component_name*, *etcd_wrapper*)

Bases: object

Intents handler class

eternal_watch(*nlu_engine*, *recursive=True*)
Watch for changes in etcd

read(*nlu_engine*, *recursive=True*, *wait=True*, *timeout=30*)
Read intent in etcd

save(*nlu_engine*)
Save intent in etcd

tuxeatpi_common.memory module

Module to handle key/value memory in Etcd

class tuxeatpi_common.memory.**MemoryHandler**(*component_name*, *etcd_wrapper*)
Bases: object

Memory handler class

delete(*key*)
Delete something in memory

read(*key*)
Read something in memory

save(*key*, *value*)
Save something in memory

tuxeatpi_common.message module

Module defining TuxEatPi Messages

class tuxeatpi_common.message.**Message**(*topic*, *data*, *context='general'*, *source=None*)
Bases: object

MQTT Message class

serialize()
Serialize message content

class tuxeatpi_common.message.**MqttClient**(*component*)
Bases: paho.mqtt.client.Client

MQTT client class

on_connect(*client*, *userdata*, *flags*, *rc*)
Callback on server connect

on_message(*mqttc*, *obj*, *msg*)
Callback on receive message

on_publish(*client*, *userdata*, *mid*)
Callback on message publish

on_subscribe(*client*, *userdata*, *mid*, *granted_qos*)
Callback on topic subcribing

run()
Run MQTT client

stop()
Stop MQTT client

class tuxeatpi_common.message.**MqttSender**(*component*)
Bases: paho.mqtt.client.Client

MQTT client class

run()
Run MQTT client

stop()
Stop MQTT client

tuxeatpi_common.message.**is_mqtt_topic**(*topic_name*)
Add a method as a MQTT topic

tuxeatpi_common.registry module

Module to handle registry in Etcd

class tuxeatpi_common.registry.**RegistryHandler**(*component_name*, *component_version*, *etcd_wrapper*)

Bases: object

Registry handler class

clear()
Remove all entries in the registry

ping(*state='ALIVE'*)
Send ping data to etcd

read()
Get all component states

set_notalive(*data*)
Set component not alive in the registry

tuxeatpi_common.settings module

Module defining how to handle component settings

class tuxeatpi_common.settings.**SettingsHandler**(*component*)

Bases: object

Settings handler class

delete(*key=None*)
Delete settings from etcd

read(*watch=False*)
Watch for component settings change in etcd

This is done by the subtaskers

read_global(*watch=False*)
Watch for global settings change in etcd

This is done by the subtaskers

save(*value*, *key=None*)
Serialize (json) value and save it in etcd

stop()
Stop waiting for settings

tuxeatpi_common.subtasker module

Module defining Base Daemon for TuxEatPi daemons

class tuxeatpi_common.subtasker.**SubTasker**(*component*)

Bases: threading.Thread

Base Daemon for TuxEatPi

run()
Startup function for main loop

stop()
Stop subtasker

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

tuxeatpi_common, 1
tuxeatpi_common.cli, 1
tuxeatpi_common.daemon, 1
tuxeatpi_common.dialogs, 2
tuxeatpi_common.error, 2
tuxeatpi_common.etcd_client, 2
tuxeatpi_common.initializer, 3
tuxeatpi_common.intents, 3
tuxeatpi_common.memory, 3
tuxeatpi_common.message, 4
tuxeatpi_common.registry, 4
tuxeatpi_common.settings, 5
tuxeatpi_common.subtasker, 5

Index

A

async_read() (tuxeatpi_common.etcd_client.EtcdWrapper method), 2

C

clear() (tuxeatpi_common.registry.RegistryHandler method), 5
cli() (in module tuxeatpi_common.cli), 1

D

delete() (tuxeatpi_common.etcd_client.EtcdWrapper method), 3
delete() (tuxeatpi_common.memory.MemoryHandler method), 3
delete() (tuxeatpi_common.settings.SettingsHandler method), 5
DialogsHandler (class in tuxeatpi_common.dialogs), 2

E

EtcdWrapper (class in tuxeatpi_common.etcd_client), 2
eternal_watch() (tuxeatpi_common.etcd_client.EtcdWrapper method), 3
eternal_watch() (tuxeatpi_common.intents.IntentsHandler method), 3

G

get_dialog() (tuxeatpi_common.daemon.TepBaseDaemon method), 1
get_dialog() (tuxeatpi_common.dialogs.DialogsHandler method), 2

H

help_() (tuxeatpi_common.daemon.TepBaseDaemon method), 1

I

Initializer (class in tuxeatpi_common.initializer), 3
IntentsHandler (class in tuxeatpi_common.intents), 3
is_mqtt_topic() (in module tuxeatpi_common.message), 4

L

load() (tuxeatpi_common.dialogs.DialogsHandler method), 2

M

main_loop() (tuxeatpi_common.daemon.TepBaseDaemon method), 1
MemoryHandler (class in tuxeatpi_common.memory), 3
Message (class in tuxeatpi_common.message), 4
MqttClient (class in tuxeatpi_common.message), 4
MqttSender (class in tuxeatpi_common.message), 4

O

on_connect() (tuxeatpi_common.message.MqttClient method), 4
on_message() (tuxeatpi_common.message.MqttClient method), 4
on_publish() (tuxeatpi_common.message.MqttClient method), 4
on_subscribe() (tuxeatpi_common.message.MqttClient method), 4

P

ping() (tuxeatpi_common.registry.RegistryHandler method), 5
publish() (tuxeatpi_common.daemon.TepBaseDaemon method), 2

R

read() (tuxeatpi_common.etcd_client.EtcdWrapper method), 3
read() (tuxeatpi_common.intents.IntentsHandler method), 3
read() (tuxeatpi_common.memory.MemoryHandler method), 3
read() (tuxeatpi_common.registry.RegistryHandler method), 5
read() (tuxeatpi_common.settings.SettingsHandler method), 5

read_global() (tuxeatpi_common.settings.SettingsHandler method), 5
RegistryHandler (class in tuxeatpi_common.registry), 4
reload() (tuxeatpi_common.daemon.TepBaseDaemon method), 2
run() (tuxeatpi_common.initializer.Initializer method), 3
run() (tuxeatpi_common.message.MqttClient method), 4
run() (tuxeatpi_common.message.MqttSender method), 4
run() (tuxeatpi_common.subtasker.SubTasker method), 5

S

save() (tuxeatpi_common.intents.IntentsHandler method), 3
save() (tuxeatpi_common.memory.MemoryHandler method), 4
save() (tuxeatpi_common.settings.SettingsHandler method), 5
serialize() (tuxeatpi_common.message.Message method), 4
set_config() (tuxeatpi_common.daemon.TepBaseDaemon method), 2
set_daemon_class() (in module tuxeatpi_common.cli), 1
set_notalive() (tuxeatpi_common.registry.RegistryHandler method), 5
SettingsHandler (class in tuxeatpi_common.settings), 5
shutdown() (tuxeatpi_common.daemon.TepBaseDaemon method), 2
signal_handler() (tuxeatpi_common.daemon.TepBaseDaemon method), 2
start() (tuxeatpi_common.daemon.TepBaseDaemon method), 2
stop() (tuxeatpi_common.message.MqttClient method), 4
stop() (tuxeatpi_common.message.MqttSender method), 4
stop() (tuxeatpi_common.settings.SettingsHandler method), 5
stop() (tuxeatpi_common.subtasker.SubTasker method), 5
SubTasker (class in tuxeatpi_common.subtasker), 5

T

TepBaseDaemon (class in tuxeatpi_common.daemon), 1
tuxeatpi_common (module), 1
tuxeatpi_common.cli (module), 1
tuxeatpi_common.daemon (module), 1
tuxeatpi_common.dialogs (module), 2
tuxeatpi_common.error (module), 2
tuxeatpi_common.etcd_client (module), 2
tuxeatpi_common.initializer (module), 3
tuxeatpi_common.intents (module), 3
tuxeatpi_common.memory (module), 3
tuxeatpi_common.message (module), 4
tuxeatpi_common.registry (module), 4
tuxeatpi_common.settings (module), 5
tuxeatpi_common.subtasker (module), 5
TuxEatPiError, 2

W

wait_for_speaking() (tuxeatpi_common.daemon.TepBaseDaemon method), 2
write() (tuxeatpi_common.etcd_client.EtcdWrapper method), 3