Pyre Documentation

Release 0.2.0

Arnaud Loonstra

Sep 05, 2017

Contents

Contents:

class pyre.**Pyre**(*name=None*, *ctx=None*, **args*, ***kwargs*)

endpoint()
Return own endpoint

events()
Iterator that yields 'PyreEvent's indefinitely

join(*group*)
Join a named group; after joining a group you can send messages to the group and all Zyre nodes in that group will receive them.

leave(*group*)
Leave a group

name()
Return our node name, after successful initialization

own_groups()
Return list of currently joined groups.

peer_address(*peer*)
Return the endpoint of a connected peer.

peer_groups()
Return list of groups known through connected peers.

peer_header_value(*peer*, *name*)
Return the value of a header of a conected peer. Returns null if peer or key doesn't exist.

peer_headers(*peer*)
Return the value of a header of a conected peer. Returns null if peer or key doesn't exist.

peers()
Return list of current peer ids.

peers_by_group(*group*)
Return list of current peer ids.

recent_events()
Iterator that yields recent 'PyreEvent's

recv()
Receive next message from network; the message may be a control message (ENTER, EXIT, JOIN, LEAVE) or data (WHISPER, SHOUT).

set_endpoint(*format*, **args*)
By default, Zyre binds to an ephemeral TCP port and broadcasts the local host name using UDP beaconing. When you call this method, Zyre will use gossip discovery instead of UDP beaconing. You MUST set-up the gossip service separately using zyre_gossip_bind() and _connect(). Note that the endpoint MUST be valid for both bind and connect operations. You can use inproc://, ipc://, or tcp:// transports (for tcp://, use an IP address that is meaningful to remote as well as local nodes). Returns 0 if the bind was successful, else -1.

set_header(*key*, *value*)
Set node header; these are provided to other nodes during discovery and come in each ENTER message.

set_interface(*value*)
Set network interface for UDP beacons. If you do not set this, CZMQ will choose an interface for you. On boxes with several interfaces you should specify which one you want to use, or strange things can happen.

set_interval(*interval*)
Set UDP beacon discovery interval, in milliseconds. Default is instant beacon exploration followed by pinging every 1,000 msecs.

set_port(*port_nbr*)
Set UDP beacon discovery port; defaults to 5670, this call overrides that so you can create independent clusters on the same network, for e.g. development vs. production. Has no effect after zyre_start().

set_verbose()
Set verbose mode; this tells the node to log all traffic as well as all major events.

shout(*group*, *msg_p*)
Send message to a named group Destroys message after sending

shouts(*group*, *format*, **args*)
Send formatted string to a named group

socket()
Return socket for talking to the Zyre node, for polling

start()
Start node, after setting header values. When you start a node it begins discovery and connection. Returns 0 if OK, -1 if it wasn't possible to start the node.

stop()
Stop node; this signals to other peers that this node will go away. This is polite; however you can also just destroy the node without stopping it.

uuid()
Return our node UUID string, after successful initialization

whisper(*peer*, *msg_p*)
Send message to single peer, specified as a UUID string Destroys message after sending

whispers(*peer*, *format*, **args*)
Send formatted string to a single peer specified as UUID string

class pyre.**PyreEvent**(*node*)
Parsing Pyre messages

This class provides a higher-level API to the Pyre.recv() call, by doing work that you will want to do in many cases, such as unpacking the peer headers for each ENTER event received.

header(*name*)
Getter for single header values

Args: name (str): Header name

Returns: str: Header value

peer_uuid
Creates uuid.UUID object

Returns: TYPE: uuid.UUID

CHAPTER 1

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index