**TopologyManager**

* add\_switch(switch\_identifier)
* remove\_switch(switch\_identifier)
* switches()
* add\_host(host\_identifier)
* update\_host(host\_identifier, \*\*host\_parameters)
* remove\_host(host\_identifier)
* hosts()
* add\_link(node\_from\_identifier, node\_to\_identifier, node\_from\_port, node\_to\_port)
* remove\_link(node\_from\_identifier, node\_to\_identifier)
* links()
* exists\_switch(switch\_identifier)
* exists\_host(host\_identifier)
* exists\_link(node\_from\_identifier, node\_to\_identifier)
* clear\_topology()
* topology()
* output\_port(source\_identifier, destination\_identifier)
* output\_port\_to\_ip\_address(source\_identifier, destination\_ip\_address)

**TopologyApplication**

Uses TopologyManager.

* event\_switch\_enter(event)
* event\_switch\_leave(event)
* event\_link\_add(event)
* event\_link\_delete(event)
* on\_packet\_in(event)
* \_\_host\_discovery(host\_identifier, ip\_address, switch\_identifier\_connected\_to, switch\_port\_connected\_to)

**REST-API**

Host: 127.0.0.1

Port: 8080

* /api/topology/ [GET]
* /api/topology/output-port/ [POST]

Example:

{

"source\_identifier": "source\_identifier",

"destination\_identifier": "destination\_identifier"

}

* /api/topology/output-port-to-ip-address/ [POST]

Example:

{

"source\_identifier": "source\_identifier",

"destination\_ip\_address": "destination\_ip\_address"

}

**FlowManager**

* send\_flow\_rules(flow\_rules)
* send\_flow\_rule(flow\_rule)
* build\_flow\_rule(switch\_identifier, priority, match, actions, idle\_timeout = 0, hard\_timeout = 0)
* build\_match(switch\_identifier, packet\_type, input\_port, match\_parameters)
* \_\_build\_match\_arp(switch\_identifier, input\_port, match\_parameters)
* \_\_build\_match\_icmp(switch\_identifier, input\_port, match\_parameters)
* \_\_build\_match\_udp(switch\_identifier, input\_port, match\_parameters)
* \_\_build\_match\_tcp(switch\_identifier, input\_port, match\_parameters)
* build\_actions(switch\_identifier, actions\_parameters)
* \_\_build\_action(switch\_identifier, action\_parameters)
* send\_groups(groups)
* send\_group(group)
* build\_add\_group(switch\_identifier, group\_type, group\_identifier, buckets)
* build\_modify\_group(switch\_identifier, group\_type, group\_identifier, buckets)
* build\_bucket(switch\_identifier, actions, weight = 1)

**FlowApplication**

* event\_switch\_enter(event)
* event\_switch\_leave(event)
* on\_packet\_in(event)
* \_\_priorities()
* \_\_output\_port(source\_identifier, destination\_identifier)
* \_\_drop\_rate(priority)
* \_\_send\_add\_group(switch\_identifier, group\_identifier, drop\_rate, priority, output\_port)
* \_\_send\_flow\_rule(switch\_identifier, input\_port, output\_port, packet\_type, match\_parameters, actions\_parameters)
* \_\_handle\_packet(message, switch\_datapath, switch\_input\_port, switch\_identifier, dl\_dst)
* \_\_flood\_packet(message, datapath, input\_port)
* \_\_send\_packet(message, datapath, input\_port, output\_port)

**REST-API**

Host: 127.0.0.1

Port: 8080

* /api/flow/send-flow-rule/ [POST]

Example:

{

"switch\_identifier": "S2",

"priority": 1,

"packet\_type": "ARP",

"input\_port": 1,

"match\_parameters": {

"eth\_src": "00:00:00:00:00:01",

"eth\_dst": "00:00:00:00:00:02"

},

"actions\_parameters": [ ["SET-QUEUE", 2], ["OUTPUT", 2] ]

}

* /api/flow/send-group/ [POST]

Example:

{

"switch\_identifier": "S2",

"group\_type": "SELECT",

"group\_identifier": 1,

"buckets": [

{

"actions": [ ["SET-QUEUE", 2], ["OUTPUT", 2] ],

"weight": 5

},

{

"actions": [ ["DROP"] ],

"weight": 5

}

]

}

* /api/flow/modify-group/ [POST]

Example:

{

"switch\_identifier": "S2",

"group\_type": "SELECT",

"group\_identifier": 1,

"buckets": [

{

"actions": [ ["SET-QUEUE", 2], ["OUTPUT", 2] ],

"weight": 2

},

{

"actions": [ ["DROP"] ],

"weight": 8

}

]

}

**FiredexMiddleware**

**REST-API**

* /api/firedex/topics/ [GET]
* /api/firedex/topic/ [POST]

Example:

{

"topic": "topic\_name"

}

* /api/firedex/priorities/ [GET]
* /api/firedex/priority/ [POST]

Example:

{

"topic": "topic\_name",

"utility\_function": "utility\_value"

}

* /api/firedex/drop-rates/ [GET]
* /api/firedex/drop-rate/ [POST]

Example:

{

"priority": "priority\_value",

"drop\_rate": "drop\_rate\_value"

}

**Publisher**

**Broker**

**Subscriber**