|  |  |
| --- | --- |
| **S\_ODLController(cmd.Cmd)**  Console interface를 위해 cmd 모듈을 상속 ( 모듈 경로 : /usr/lib/python2.7)  Controller에 관련된 명령들을 수행  입력된 문자열에 해당하는 명령을 수행 | |
| Method  LIst | **C\_Start(ip, port)**  입력 받은 인자를 저장.  username, password를 입력 받아, odl\_api 클래스를 생성  Console interface를 시작 |
| **do\_show(command)**  console에 show <command> 형식의 입력을 처리  정보 출력에 관련 된 명령을 실행   |  |  | | --- | --- | | 커맨드 목록 | | | switches | 스위치들의 정보를 출력 | | hosts | 스위치에 연결된 호스트들의 정보를 출력 | | links | 스위치와 호스트들이 연결 된 정보를 출력 | | flows | 모든 flow 정보를 출력 | | controller\_info | 현재 컨트롤러의 정보를 출력 | |
| **do\_switch(switch\_name)**  console에 switch <switch\_name> 형식의 입력을 처리  입력 된 switch\_name이 존재하는지 확인  입력 된 switch\_name에 해당하는 스위치의 console을 실행  username, password를 입력 받아 api를 사용하기 위한 token 생성 |
| **do\_alias(controller\_name)**  controller를 특정 지을 수 있는(Unique) 이름을 입력 받아 현재 컨트롤러의 이름으로 사용 |
| **do\_help()**  controller console의 입력형식, 명령어 등 사용법을 출력 |
| **complete\_show(text, line, begidx, endidx)**  show command에 대한 help와 option을 출력 |
| **complete\_switch(text, line, begidx, endidx)**  flow command에 대한 help를 출력 |
| **complete\_alias(text, line, begidx, endidx)**  alias command에 대한 help를 출력 |

|  |  |
| --- | --- |
| **S\_ODLSwtich(cmd.Cmd)**  Console interface를 위해 cmd 모듈을 상속 ( 모듈 경로 : /usr/lib/python2.7)  Switch에 관련된 명령들을 수행  입력된 문자열에 해당하는 명령을 수행 | |
| Method  List | **S\_Start(ip, port, dpid, token)**  입력 받은 인자를 저장  Console interface를 시작 |
| **do\_show(command)**  console에 show <command> 형식의 입력을 처리  정보 출력에 관련 된 명령을 실행   |  |  | | --- | --- | | 커맨드 목록 | | | ports | 스위치의 포트 정보를 출력 | | flows | 스위치의 flow 정보를 출력 | |
| **do\_flow(command)**  console에 flow <commad> 형식의 입력을 처리  flow를 설정할 수 있는 모듈을 실행 # TODO |
| **do\_alias(switch\_name)**  switch를 특정 지을 수 있는(Unique) 이름을 입력 받아 현재 switch의 이름으로 사용 |
| **do\_help()**  switch console의 입력형식, 명령어 등 사용법을 출력 |
| **complete\_show(text, line, begidx, endidx)**  show command에 대한 help와 option을 출력 |
| **complete\_flow(text, line, begidx, endidx)**  flow command에 대한 help를 출력 |
| **complete\_alias(text, line, begidx, endidx)**  alias command에 대한 help를 출력 |

|  |  |
| --- | --- |
| **S\_ODLFlow(cmd.Cmd)**  Console interface를 위해 cmd 모듈을 상속 ( 모듈 경로 : /usr/lib/python2.7)  flow에 관련된 명령들을 수행  입력된 문자열에 해당하는 명령을 수행 | |
| Method  List | **F\_Start(**ip, port, dpid, token**)**  입력 받은 인자를 저장  Console interface를 시작 |
| **do\_quit(), do\_exit()**  cmd를 종료 |
| **do\_help()**  flow cmd의 명령어, 사용 방법을 출력. |
| **do\_alias()**  flow를 특정할 수 있는 이름을 입력 받아 저장 |
| **do\_show()**  console에 show 명령을 처리,  현재까지의 생성 한 flow 정보를 출력 |
| **do\_match(match, match\_value)**  openflow의 match에 해당하는 값을 받아 저장 |
| **do\_instruction(instruction, instruction\_value)**  openflow의 instruction에 해당하는 값을 받아 저장 |
| **do\_save\_flow()**  현재까지 생성 된 flow 정보를 파일로 저장 |
| **do\_send\_flow()**  현재까지 생성 된 flow 정보를 이용해 controller의 RestAPI 호출 |
| **do\_priority(priority)**  priority를 입력 받아 저장 |
| **complete\_show(text, line, begidx, endidx)**  show command에 대한 help와 option을 출력 |
| **complete\_match(text, line, begidx, endidx)**  match command에 대한 help와 option을 출력 |
| **complete\_instruction(text, line, begidx, endidx)**  instruction command에 대한 help와 option을 출력 |
| **complete\_save\_flow(text, line, begidx, endidx)**  save\_flow command에 대한 help와 option을 출력 |
| **complete\_send\_flow(text, line, begidx, endidx)**  send\_flow command에 대한 help와 option을 출력 |
| **complete\_priority(text, line, begidx, endidx)**  priority command에 대한 help와 option을 출력 |

|  |  |
| --- | --- |
| **odl\_api()**  ODL의 RestAPI를 call 하여 결과 값을 처리 | |
| Method  List | **login(host, username, password)**  username, password를 이용하여 ODL\_Auth API를 이용해 token을 리턴  retrun token |
| **get\_connection(host, token)**  token과 ip, port를 이용해 HTTP 커넥션을 생성. |
| **controller\_show\_switches(host, token)**  controller가 관리하는 모든 스위치를 출력 |
| **controller\_show\_hosts(host, token)**  controller가 관리하는 모든 스위치에 붙어있는 모든 호스트를 출력 |
| **controller\_show\_links(host, token)**  controller가 관리하는 모든 스위치와 호스트의 연결 정보를 출력 |
| **controller\_show\_flows(host, token)**  controller가 관리하는 모든 플로우를 출력 |
| **switch\_show\_ports(host, token, dpid)**  입력된 dpid에 해당하는 스위치의 포트 정보를 출력 |
| **switch\_show\_flow(host, token, dpid, command)**  dpid에 해당하는 스위치의 flow 정보를 출력 |
| **flow\_add(dpid, token, flow\_info)**  controller에 입력된 flow 정보를 추가하는 RestAPI를 요청 |
| **flow\_delete(dpid, token, flow\_info)**  controller에 입력된 flow 정보를 삭제하는 RestAPI를 요청 |
| **flow\_update(dpid, token, flow\_info)**  controller에 입력된 flow 정보를 수정하는 RestAPI를 요청 |
| Variable List | **URI\_TOKEN**  Token을 생성하는 RestAPI의 URI  = "/oauth2/token" |
| **URI\_ALL\_SWITCHES**  모든 스위치의 정보를 얻어오는 RestAPI의 URI  = /restconf/operational/opendaylight-inventory:nodes/ |
| **URI\_ALL\_LINKS**  모든 링크 정보를 얻어오는 RestAPI의 URI  = /restconf/operational/network-topology:network-topology/ |
| **URI\_ALL\_FLOWS**  모든 flow 정보를 얻어오는 RestAPI의 URI  = /restconf/operational/opendaylight-inventory:nodes/ |
| **URI\_SWITCH\_PORT**  스위치의 포트정보를 얻어오는 RestAPI의 URI |
| **URI\_SWITCH\_FLOW**  스위치의 flow 정보를 얻어오는 RestAPI의 URI |
| **HEADERS\_REQUEST**  RestAPI를 Request 할 때 필요한 HTTP Header |
| **HEADERS\_GET\_TOKEN**  Token을 생성하는 RestAPI를 Request 할 때 필요한 HTTP Header |
| **PARAMS\_GET\_TOKEN**  Token을 생성하는 RestAPI를 Request 할 때 필요한 x-www-form-urlencorded parameter |
| **MATCHES**  openflow의 match list  [ eth\_src, eth\_dst, ipv4\_src, ipv4\_dst, eth\_type, protocol, srcport, dstport, vlanid, vlanpriority, tos, apply ] |