Python	Restful	Chức năng	Đầu ra
homeassistant.remote.get_config(api)	GET /api/config	Return configuration	{"components": ["automation","logbook","websocket_api","updater","config.core","config.automation","frontend","mqtt","conversation","history","api","config.group","sensor","binary_sensor","light.mqtt","sensor.mqtt","config","http","recorder","light","discovery","group","binary_sensor.mqtt"],"config_dir": "/config","elevation": 5,"latitude": 10.8142,"location_name": "Home","longitude": 106.6438,"time_zone": "Asia/Ho_Chi_Minh","unit_system": {"length": "km","mass": "g","temperature": "°C","volume": "L"},"version": "0.48.1","whitelist_external_dirs": ["/config/www"]}
	GET /api/discovery_info	Returns basic information about the Home Assistant instance as JSON	{"base_url": "http://172.17.0.3:8123","location_name": "Home","requires_api_password": false,"version": "0.48.1"}
homeassistant.remote.get_event_listeners(api)	GET /api/events	Returns an array of event objects. Each event object contains event name and listener count.	[{"event": "homeassistant_close","listener_count": 3},{"event": "call_service","listener_count": 1},{"event": "homeassistant_stop","listener_count": 5},{"event": "*","listener_count": 1},{"event": "time_changed","listener_count": 2},{"event": "component_loaded","listener_count": 2},{"event": "platform_discovered","listener_count": 3},{"event": "state_changed","listener_count": 4},{"event": "service_registered","listen

			er_count": 1},{"event": "service_removed","listener_count": 1}]
homeassistant.remote.get_services(api)	GET /api/services	Returns an array of service objects. Each object contains the domain and which services it contains.	[...{"domain": "mqtt","services": {"publish": {"description": "Publish a message to an MQTT topic","fields": {"payload": {"description": "Payload to publish","example": "This is great"},"payload_template": {"description": "Template to render as payload value. Ignored if payload given.","example": "{{ states('sensor.temperature') }}"},"qos": {"default": 0,"description": "Quality of Service","example": 2,"values": [0,1,2]},"retain": {"default": false,"description": "If message should have the retain flag set.","example": true},"topic": {"description": "Topic to publish payload","example": "/homeassistant/hello"}}}}},....]
	GET /api/history/period/<timestamp> filter_entity_id=<entity_id> to filter on a single entity end_time=<timestamp> to choose the end of the period in URL encoded format (defaults to 1 day).	Returns an array of state changes in the past. Each object contains further details for the entities	[..[{"attributes": {"device_class": "motion","friendly_name": "Motion Detection"},"entity_id": "binary_sensor.motion_detection","last_changed": "2018-01-17T14:54:28.272256+00:00","last_updated": "2018-01-17T14:54:28.272256+00:00","state": "off"},{"attributes": {"device_class": "motion","friendly_name": "Motion Detection"},"entity_id": "binary_sensor.motion_detection","last_changed": "2018-01-17T15:26:49.125524+00:00","last_update

			d": "2018-01-17T15:26:49.125524+00:00","state": "off"}],...]
homeassistant.remote.get_states(api)	GET /api/states	Returns an array of state objects. Each state has the following attributes: entity_id, state, last_changed and attributes.	[{"attributes": {"device_class": "motion","friendly_name": "Motion Detection"},"entity_id": "binary_sensor.motion_detection","last_changed": "2018-01-17T15:26:49.125524+00:00","last_updated": "2018-01-17T15:26:49.125524+00:00","state": "off"},{"attributes": {"friendly_name": "Humidity","unit_of_measurement": "%"},"entity_id": "sensor.humidity","last_changed": "2018-01-17T15:26:49.136202+00:00","last_updated": "2018-01-17T15:26:49.136202+00:00","state": "unknown"},{"attributes": {"unit_of_measurement": "°C"},"entity_id": "sensor.temperature","last_changed": "2018-01-18T02:59:27.287829+00:00","last_updated": "2018-01-18T02:59:27.287829+00:00","state": "225"}]
homeassistant.remote.get_state(api, entity_id)	GET /api/states/<entity_id>	Returns a state object for specified entity_id	{"attributes": {},"entity_id": "light.red_light","last_changed": "2018-01-18T02:56:48.286098+00:00","last_updated": "2018-01-18T02:56:48.286098+00:00","state": "OFF"}
	GET /api/error_log	Retrieve all errors logged during the current session of Home Assistant as a	File log các lỗi trong quá trình chạy

		plaintext response.	
homeassistant.remote.set_state(api, entity_id, new_state, attributes=None, force_update=False)	POST /api/states/<entity_id>	Updates or creates the current state of an entity.	
homeassistant.remote.call_service(api, domain, service, service_data=None, timeout=5)	POST /api/services/<domain>/<service>	Calls a service within a specific domain	
homeassistant.remote.fire_event(api, event_type, data=None)	POST /api/events/<event_type>	Fires an event with event_type	
homeassistant.remote.remove_state(api, entity_id)		Call API to remove state for entity_id	