
Switcher Webapi

Release 0.5.0

Tomer Figenblat

May 17, 2020

CONTENTS

1	Install	1
2	Prerequisites	3
3	Notes	5
4	Usage	7
5	License	15
6	Credits	17
7	Code Documentation	19
	Python Module Index	29
	Index	31

INSTALL

```
docker run -d -p 8000:8000 \  
-e CONF_DEVICE_IP_ADDR=192.168.100.157 \  
-e CONF_PHONE_ID=1234 \  
-e CONF_DEVICE_ID=ab1c2d \  
-e CONF_DEVICE_PASSWORD=12345678 \  
--name switcher_webapi tomerfi/switcher_webapi:latest"
```

You can also add another optional environment variable:

```
-e CONF_THROTTLE=5.0
```

for setting the throttle time between consecutive requests, this is optional and the default value is **5.0**.

Here's an example of running the container using *docker-compose* setting the environment variables in a designated file.

```
# docker-compose.yml  
version: "3.7"  
  
services:  
  switcher_api:  
    image: tomerfi/switcher_webapi:latest  
    container_name: "switcher_webapi"  
    env_file:  
      - .env_vars  
    ports:  
      - 8000:8000  
    restart: unless-stopped
```

```
# .env_vars  
CONF_DEVICE_IP_ADDR=192.168.100.157  
CONF_PHONE_ID=1234  
CONF_DEVICE_ID=ab1c2d  
CONF_DEVICE_PASSWORD=12345678  
CONF_THROTTLE=5.0
```


PREREQUISITES

- Install and configure your Switcher device.
- **Collect the following information from the device's following NightRang3r instructions in the [Switcher-V2-Python](#) repository**
 - ip_address
 - phone_id
 - device_id
 - device_pass
- Install docker

NOTES

- If you don't want to be forced to restart the container if the device's ip address changes, please consider assigning the device with a static ip address.
- The Switcher-V2-Python repository is build with python 2.7.
- The [aioswitcher](#) was tested with the Switcher V2 device by myself and with the Switcher Touch device by the community.
- This project was intended for local usage, it's ok if you want to use it remotely, just make sure to take the proper security measures such as reverse proxy and ssl.
- The WebAPI has a throttle mechanism to prevent overflowing the device with frequent requests, it defaults to 5 seconds throttle time.
- Some users have been reporting lately about failures using the Switcher-V2-Python script after upgrading the device firmware to 3.0, please follow the relevant issues in the script repository before doing the same.

USAGE

Once running, you can send REST requests towards the container. With the exception of the *create_schedule* requests, all the requests requiring input accepts it as a json body or in the form of query parameters.

4.1 get_state

URL: */switcher/get_state*

Method: *GET*

Request parameters: *None*

Response body example:

```
{
  "successful": true,
  "state": "on",
  "time_left": "00:47:25",
  "time_on": "00:12:35",
  "auto_off": "01:30:00",
  "power_consumption": 2669,
  "electric_current": 12.3
}
```

4.2 turn_on

URL: */switcher/turn_on*

Method: *POST*

Request parameters:

Key	Required	Description
minutes	<i>Optional</i>	turn on the device with an off timer of 1-180 minutes.

Request body example:

```
{
  "minutes": "30"
}
```

Response body example:

```
{  
  "successful": true  
}
```

4.3 turn_off

URL: */switcher/turn_off*

Method: *POST*

Request parameters: *None*

Response body example:

```
{  
  "successful": true  
}
```

4.4 set_auto_shutdown

URL: */switcher/set_auto_shutdown*

Method: *POST*

Request parameters:

Key	Required	Description
hours	<i>Mandatory</i>	hours value 1-3.
minutes	<i>Mandatory</i>	minutes value 0-59.

Note: The auto shutdown configuration value accept any total value of hours and minutes between 1 and 3 hours.

Request body example:

```
{  
  "hours": "1",  
  "minutes": "30"  
}
```

Response body example:

```
{  
  "successful": true  
}
```

4.5 set_device_name

URL: */switcher/set_device_name*

Method: *POST*

Request parameters:

Key	Required	Description
name	<i>Mandatory</i>	device name, accepts length of 2-32 characters.

Request body example:

```
{
  "name": "my new device name"
}
```

Response body example:

```
{
  "successful": true
}
```

4.6 get_schedules

URL: */switcher/get_schedules*

Method: *GET*

Request parameters: *None*

Response body example:

```
{
  "successful": true,
  "found_schedules": true,
  "schedules": [
    {
      "schedule_id": "0",
      "enabled": true,
      "recurring": true,
      "days": [
        "Tuesday",
        "Wednesday",
        "Thursday",
        "Friday",
        "Saturday",
        "Sunday"
      ],
      "start_time": "17:30",
      "end_time": "18:30",
      "duration": "1:00:00",
      "schedule_data": "0001fc01e871a35cf87fa35c",
      "next_run": "Due next Tuesday at 17:30"
    },
  ],
}
```

(continues on next page)

(continued from previous page)

```

{
  "schedule_id": "1",
  "enabled": true,
  "recurring": true,
  "days": ["Monday"],
  "start_time": "17:00",
  "end_time": "18:00",
  "duration": "1:00:00",
  "schedule_data": "0101020160a6c95c70b4c95c",
  "next_run": "Due tommorow at 17:00"
}
]
}

```

Note: The *schedules* list can contain up to 8 schedules with the identifiers of 0-7 representing the actual schedule slots on the device.

4.7 enable_schedule

URL: */switcher/enable_schedule*

Method: *PATCH*

Request parameters:

Key	Re-quired	Description
schedule_data	<i>Mandatory</i>	the <i>schedule_data</i> associated with the chosen schedule. retrieved with <i>/switcher/get_schedules</i> .

Request body example:

```

{
  "schedule_data": "0101020160a6c95c70b4c95c"
}

```

Response body example:

```

{
  "successful": true
}

```

4.8 disable_schedule

URL: */switcher/disable_schedule*

Method: *PATCH*

Request parameters:

Key	Re-quired	Description
schedule_data	<i>Mandatory</i>	the <i>schedule_data</i> associated with the chosen schedule. retrieved with <i>/switcher/get_schedules</i> .

Request body example:

```
{
  "schedule_data": "0101020160a6c95c70b4c95c"
}
```

Response body example:

```
{
  "successful": true
}
```

4.9 delete_schedule

URL: */switcher/delete_schedule*

Method: *DELETE*

Request parameters:

Key	Required	Description
schedule_id	<i>Mandatory</i>	the <i>schedule_id</i> associated with the chosen schedule. retrieved with <i>/switcher/get_schedules</i> .

Request body example:

```
{
  "schedule_id": "2"
}
```

Response body example:

```
{
  "successful": true
}
```

4.10 create_schedule

URL: `/switcher/create_schedule`

Method: `PUT`

Request parameters:

Key	Required	Description
days	<i>Mandatory</i>	list of days for the schedule to run in. (empty for non-recurring schedules).
start_hours	<i>Mandatory</i>	start time hours value 0-23.
start_minutes	<i>Mandatory</i>	start minutes value 0-59.
stop_hours	<i>Mandatory</i>	stop time hours value 0-23.
stop_minutes	<i>Mandatory</i>	stop minutes value 0-59.

Request body example:

```
{
  "days": ["Monday", "Wednesday", "Friday"],
  "start_hours": "20",
  "start_minutes": "30",
  "stop_hours": "21",
  "stop_minutes": "0"
}
```

Response body example:

```
{
  "successful": true
}
```

Possible values for the *days* list:

- Sunday
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

Note: Due to its complexity, the *create_schedule* request accepts its arguments in the form of a json body only, query parameters will not be accepted.

4.11 Exceptions

Unless unhandled, all exceptions will return a json object in response body:

```
{  
  "successful": false,  
  "message": "the error description"  
}
```


LICENSE

MIT License

Copyright © 2019 Tomer Figenblat

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CREDITS

This project was enabled by creating the [aioswitcher](#) pypi module, initially created for use with the [home assistant component](#).

Not this nor the aioswitcher project would have been able to happen without the amazing work preformed by Nigh-tRang3r and AviadGolan in the [Switcher-V2-Python](#) project.

So... Thanks!

CODE DOCUMENTATION

7.1 Application run scripts

7.1.1 pyscripts/request_handlers.py

Request handlers for the Switcher WebAPI.

```
async request_handlers._create_raw_schedule_data (schedule_days: List[int],  
start_hours: int, start_minutes:  
int, stop_hours: int, stop_minutes:  
int) → str
```

Use as helper creating raw schedule data for creating schedules.

Parameters

- **schedule_days** – selected days for the schedule to run in.
- **start_hours** – hour to start the device at.
- **start_minutes** – minutes to start the device at.
- **stop_hours** – hour to stop the device at.
- **stop_minutes** – minutes to stop the device at.

Returns Raw schedule data needed for creating the requested schedule.

```
async request_handlers._parse_schedule_body (body: Dict) → str
```

Use as helper parsing body of create schedule requests.

Parameters **body** – json body of the create schedule requests.

Raises **sanic.exceptions.InvalidUsage** – when missing a mandatory argument.

Returns Schedule data object needed for creating the new schedules.

```
request_handlers._validate_day_to_int (day: str) → int
```

Use as helper converting string weekday to int for creating schedules.

Parameters **day** – string representation of the weekday.

Raises **sanic.exceptions.InvalidUsage** – when encountered unknown weekday string.

Returns

The int representation of the weekday.

More information is available in the Usage section.

`request_handlers._validate_time_integers` (*start_hours: int, start_minutes: int, stop_hours: int, stop_minutes: int*) → None

Use as helper validating time arguments of creating schedule requests.

Parameters

- **start_hours** – hour to start the device at (0-23).
- **start_minutes** – minutes to start the device at (0-59).
- **stop_hours** – hour to stop the device at (0-23).
- **stop_minutes** – minutes to stop the device at (0-59).

Raises `sanic.exceptions.InvalidUsage` – when the validation fails.

async `request_handlers.create_schedule_handler` (*request: sanic.request.Request, ip_address: str, phone_id: str, device_id: str, device_password: str*) → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/create_schedule`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises `sanic.exceptions.ServerError` – when encountered an error.

Returns

Json object representing the request status.

More information is available in the `Usage` section.

Warning: Accepts json body only, no query parameters allowed.

async `request_handlers.delete_schedule_handler` (*request: sanic.request.Request, ip_address: str, phone_id: str, device_id: str, device_password: str*) → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/delete_schedule`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises

- `sanic.exceptions.InvalidUsage` – when encountered unknown weekday.

- `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.disable_schedule_handler` (*request:* `sanic.request.Request`,
ip_address: `str`, *phone_id:* `str`, *device_id:* `str`, *device_password:* `str`) →
`sanic.response.HTTPResponse`

Use for handling requests to `/switcher/disable_schedule`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises

- `sanic.exceptions.InvalidUsage` – when encountered faulty data.
- `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.enable_schedule_handler` (*request:* `sanic.request.Request`,
ip_address: `str`, *phone_id:* `str`, *device_id:* `str`, *device_password:* `str`) →
`sanic.response.HTTPResponse`

Use for handling requests to `/switcher/enable_schedule`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises

- `sanic.exceptions.InvalidUsage` – when encountered faulty data.

- `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.get_schedules_handler` (*request: `sanic.request.Request`,
ip_address: str, phone_id: str, device_id: str, device_password: str*) → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/get_schedules`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the configured schedules on the device.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.get_state_handler` (*request: `sanic.request.Request`, ip_address: str,
phone_id: str, device_id: str, device_password: str*) → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/get_state`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the current state of the device.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.set_auto_shutdown_handler` (*request:* `sanic.request.Request`,
ip_address: `str`, *phone_id:* `str`, *device_id:* `str`, *device_password:* `str`)
 → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/set_auto_shutdown`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises

- `sanic.exceptions.InvalidUsage` – when requested is not 59-180 minutes.
- `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the `Usage` section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.set_device_name_handler` (*request:* `sanic.request.Request`,
ip_address: `str`, *phone_id:* `str`, *device_id:* `str`, *device_password:* `str`) →
`sanic.response.HTTPResponse`

Use for handling requests to `/switcher/set_device_name`.

Parameters

- **request** – `sanic`'s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises

- `sanic.exceptions.InvalidUsage` – when name length is no 2-32 characters.
- `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the `Usage` section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.turn_off_handler` (*request: sanic.request.Request, ip_address: str, phone_id: str, device_id: str, device_password: str*) → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/turn_off`.

Parameters

- **request** – sanic’s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

async `request_handlers.turn_on_handler` (*request: sanic.request.Request, ip_address: str, phone_id: str, device_id: str, device_password: str*) → `sanic.response.HTTPResponse`

Use for handling requests to `/switcher/turn_on`.

Parameters

- **request** – sanic’s request object.
- **ip_address** – the local ip address.
- **phone_id** – the extracted phone id.
- **device_id** – the extracted device id.
- **device_password** – the extracted device password.

Raises

- `sanic.exceptions.InvalidUsage` – when timer is no 1-180 minutes.
- `sanic.exceptions.ServerError` – when encountered any error.

Returns

Json object representing the request status.

More information is available in the Usage section.

Note: Accepts arguments as json body or query parameters.

7.1.2 pyscripts/start_server.py

Sanic server for the Switcher WebAPI.

`start_server.before_start` (*app*: `sanic.Sanic`, *loop*: `asyncio.events.AbstractEventLoop`) → None
Use for preparing data and register mappings before start.

This function is annotated with `sanic.Sanic.listener("before_server_start")`.

It is called by `Sanic` just before the server starts. Its job is:

- Gather the initial data for running the server.
- Register a middleware for acquiring a throttler for all requests.
- Add routes using the `mappings` module and the `request_handlers` module.

Parameters

- **app** – the running `sanic` app.
- **loop** – the main event loop.

`start_server.timeout` (*request*: `sanic.request.Request`, *exception*: `sanic.exceptions.SanicException`)
→ `sanic.response.HTTPResponse`
Use as custom handler for logging internal service errors.

This function is annotated with `sanic.Sanic.exception(ServerError)`.

It is called by `Sanic` for every `ServerError` exception.

Its job is to log the exception and return a code 500 response.

Parameters

- **request** – the incoming request object.
- **exception** – the exception thrown.

7.2 Unit testing scripts

7.2.1 pyscripts/confstest.py

Fixtures and mockings for unit testing the Switcher WebAPI.

`confstest.mock_control_response` () → `Generator[asyncmock.MagicMock, Any, None]`
Fixture for mocking the control response.

Yields Mocked `SwitcherV2ControlResponseMSG` object.

`confstest.mock_create_schedule_request` () → `Generator[asyncmock.MagicMock, Any, None]`
Fixture for mocking the create_schedule response.

Yields Mocked `SwitcherV2CreateScheduleResponseMSG` object.

`confstest.mock_delete_schedule_request` () → `Generator[asyncmock.MagicMock, Any, None]`
Fixture for mocking the delete_schedule response.

Yields Mocked `SwitcherV2DeleteScheduleResponseMSG` object.

`confstest.mock_disable_enable_schedule_request` () → `Generator[asyncmock.MagicMock, Any, None]`
Fixture for mocking the disable_enable_schedule response.

Yields Mocked `SwitcherV2DisableEnableScheduleResponseMSG` object.

`confptest.mock_get_schedules_response` (*schedule_object*) → Generator[`asyncstest.MagicMock`,
Any, None]

Fixture for mocking the `get_schedules` response.

Parameters `schedule_object` – Fixture of mocked `SwitcherV2Schedule` object.

Yields Mocked `SwitcherV2GetScheduleResponseMSG` object.

`confptest.mock_get_state_response` () → Generator[`asyncstest.MagicMock`, Any, None]

Fixture for mocking the `get_state` response.

Yields Mocked `SwitcherV2StateResponseMSG` object.

`confptest.mock_loop` () → Generator[`asyncio.events.AbstractEventLoop`, Any, None]

Fixture for running an event loop.

Yields Test event loop for running test server.

`confptest.mock_sanica_test_app` () → Generator[`sanica.Sanica`, Any, None]

Fixture for creating a test instance on the sanica app.

Yields Test sanica application for mocking testing server.

`confptest.mock_schedule_object` () → Generator[None, None,
`aioswitcher.schedules.SwitcherV2Schedule`]

Fixture for the `aioswitcher.schedules.SwitcherV2Schedule` object.

Returns Mocked `SwitcherV2Schedule` object.

`confptest.mock_set_auto_shutdown_response` () → Generator[`asyncstest.MagicMock`, Any,
None]

Fixture for mocking the `set_auto_shutdown` response.

Yields Mocked `SwitcherV2SetAutoOffResponseMSG` object.

`confptest.mock_set_device_name_response` () → Generator[`asyncstest.MagicMock`, Any, None]

Fixture for mocking the `set_device_name` response.

Yields Mocked `SwitcherV2UpdateNameResponseMSG` object.

`confptest.mock_switcher_api_context_manager` () → Generator[None, Any, None]

Fixture for mocking the `SwitcherV2Api` context manager.

`confptest.mock_tcp_connection` () → Generator[None, Any, None]

Fixture for mocking `asyncio.open_connection`.

`confptest.mock_test_client` (*loop: asyncio.events.AbstractEventLoop*, *sanica_test_app: sanica.Sanica*)
→ Generator[None, None, `asyncio.events.AbstractEventLoop`]

Fixture for starting server in the event loop.

Parameters

- `loop` – Fixture of mocked `AbstractEventLoop` object.
- `sanica_test_app` – Fixture of mocked `Sanica` object.

Returns An event loop with a running server.

7.2.2 pyscripts/helpers.py

Helper functions for unit testing the Switcher WebAPI.

`helpers.get_local_ip_address()` → str

Use for getting the local host's ip address.

Returns The local ip address.

`helpers.get_next_weekday(is_iso: bool = False)` → int

Use for getting next day weekday.

Parameters `is_iso` – If true, Monday=1 and Sunday=7. Else Monday=0 and Sunday=6.

Returns The int value representing the the next weekday (tomorrow).

7.2.3 pyscripts/test_server.py

Unit tests for the Switcher WebAPI.

async `test_server.test_create_schedule_request` (`create_schedule_response: async_test.MagicMock`) → None

Unit test-cases for /switcher/create_schedule request.

Parameters `create_schedule_response` – fixture of mocked SwitcherV2CreateScheduleResponseMSG object.

async `test_server.test_delete_schedule_request` (`delete_schedule_response: async_test.MagicMock`) → None

Unit test-cases for /switcher/delete_schedule request.

Parameters `delete_schedule_response` – fixture of mocked SwitcherV2DeleteScheduleResponseMSG object.

async `test_server.test_disable_schedule_request` (`disable_enable_schedule_response: async_test.MagicMock`) → None

Unit test-cases for /switcher/disable_schedule request.

Parameters `disable_enable_schedule_response` – fixture of mocked SwitcherV2DisableEnableScheduleResponseMSG object.

async `test_server.test_enable_schedule_request` (`disable_enable_schedule_response: async_test.MagicMock`) → None

Unit test-cases for /switcher/enable_schedule request.

Parameters `disable_enable_schedule_response` – fixture of mocked SwitcherV2DisableEnableScheduleResponseMSG object.

async `test_server.test_get_schedules_request` (`get_schedules_response: async_test.MagicMock`) → None

Unit test-cases for /switcher/get_schedules request.

Parameters `get_schedules_response` – fixture of mocked SwitcherV2GetScheduleResponseMSG object.

async `test_server.test_get_state_request` (`get_state_response: async_test.MagicMock`) → None

Unit test-cases for /switcher/get_state request.

Parameters `get_state_response` – fixture of mocked SwitcherV2StateResponseMSG object.

async `test_server.test_set_auto_shutdown_request` (*set_auto_shutdown_response: asynctest.MagicMock*) → None

Unit test-cases for /switcher/set_auto_shutdown request.

Parameters `set_auto_shutdown_response` – fixture of mocked `SwitcherV2SetAutoOffResponseMSG` object.

async `test_server.test_set_device_name_request` (*set_device_name_response: asynctest.MagicMock*) → None

Unit test-cases for /switcher/set_device_name request.

Parameters `set_device_name_response` – fixture of mocked `SwitcherV2UpdateNameResponseMSG` object.

async `test_server.test_turn_off_request` (*control_response: asynctest.MagicMock*) → None

Unit test-cases for /switcher/turn_off request.

Parameters `control_response` – fixture of mocked `SwitcherV2ControlResponseMSG` object.

async `test_server.test_turn_on_request` (*control_response: asynctest.MagicMock*) → None

Unit test-cases for /switcher/turn_on request.

Parameters `control_response` – fixture of mocked `SwitcherV2ControlResponseMSG` object.

7.3 Shared scripts

7.3.1 pyscripts/consts.py

Various constants and test values for the Switcher WebAPI project.

7.3.2 pyscripts/mappings.py

Url mappings for the Switcher WebAPI project are located here.

PYTHON MODULE INDEX

c

conftest, 25

consts, 28

h

helpers, 27

m

mappings, 28

r

request_handlers, 19

s

start_server, 25

t

test_server, 27

Symbols

`_create_raw_schedule_data()` (in module *request_handlers*), 19
`_parse_schedule_body()` (in module *request_handlers*), 19
`_validate_day_to_int()` (in module *request_handlers*), 19
`_validate_time_integers()` (in module *request_handlers*), 19

B

`before_start()` (in module *start_server*), 25

C

conftest
 module, 25
consts
 module, 28
`create_schedule_handler()` (in module *request_handlers*), 20

D

`delete_schedule_handler()` (in module *request_handlers*), 20
`disable_schedule_handler()` (in module *request_handlers*), 21

E

`enable_schedule_handler()` (in module *request_handlers*), 21

G

`get_local_ip_address()` (in module *helpers*), 27
`get_next_weekday()` (in module *helpers*), 27
`get_schedules_handler()` (in module *request_handlers*), 22
`get_state_handler()` (in module *request_handlers*), 22

H

helpers

module, 27

M

mappings
 module, 28
`mock_control_response()` (in module *conftest*), 25
`mock_create_schedule_request()` (in module *conftest*), 25
`mock_delete_schedule_request()` (in module *conftest*), 25
`mock_disable_enable_schedule_request()` (in module *conftest*), 25
`mock_get_schedules_response()` (in module *conftest*), 26
`mock_get_state_response()` (in module *conftest*), 26
`mock_loop()` (in module *conftest*), 26
`mock_sanit_test_app()` (in module *conftest*), 26
`mock_schedule_object()` (in module *conftest*), 26
`mock_set_auto_shutdown_response()` (in module *conftest*), 26
`mock_set_device_name_response()` (in module *conftest*), 26
`mock_switcher_api_context_manager()` (in module *conftest*), 26
`mock_tcp_connection()` (in module *conftest*), 26
`mock_test_client()` (in module *conftest*), 26
module
conftest, 25
consts, 28
helpers, 27
mappings, 28
request_handlers, 19
start_server, 25
test_server, 27

R

request_handlers
 module, 19

S

`set_auto_shutdown_handler()` (in module *re-*

quest_handlers), 23
set_device_name_handler() (in module *request_handlers*), 23
start_server
 module, 25

T

test_create_schedule_request() (in module *test_server*), 27
test_delete_schedule_request() (in module *test_server*), 27
test_disable_schedule_request() (in module *test_server*), 27
test_enable_schedule_request() (in module *test_server*), 27
test_get_schedules_request() (in module *test_server*), 27
test_get_state_request() (in module *test_server*), 27
test_server
 module, 27
test_set_auto_shutdown_request() (in module *test_server*), 27
test_set_device_name_request() (in module *test_server*), 28
test_turn_off_request() (in module *test_server*), 28
test_turn_on_request() (in module *test_server*), 28
timeout() (in module *start_server*), 25
turn_off_handler() (in module *request_handlers*), 24
turn_on_handler() (in module *request_handlers*), 24