Shelly Family Overview

These pages describe the HTTP API exposed by the Shelly family of devices.

Devices in the Shelly family are IoT nodes connected to the Internet over WiFi. All devices support a common set of configuration parameters, some share common features. Apart from these, each device extends the common HTTP endpoints with a set of device-specific settings and behavior.

WiFi Modes

Shelly devices can operate as either WiFi Access Point (AP mode) or Client Mode (STA). The factory default is AP mode with SSID `shelly<MODEL>-XXXXXXXXXXXX` with no authentication enabled.

Initially, devices come preprogrammed in Access Point mode with no password set. To be able to connect to Allterco's cloud service, synchronize time, etc. the device has to be configured to connect to an existing, Internet-connected WiFi infrastructure.

Connecting the device to an existing WiFi infrastructure can be done:

- Via Allterco's mobile applications for Android and iOS
- Using the local web interface via a browser (open http://192/168.33.1/)
- By performing a HTTP request to set the desired WiFi settings (see HTTP API below).

Control of these parameters via HTTP is possible via the `/settings/sta` and `/settings/ap` endpoints.

HTTP Server

All devices run a local HTTP server on port 80. It serves as a simple web page which allows the user to setup basic parameters. While in AP WiFi mode, the web page can be accessed at:

http://192.168.33.1/

The web interface makes use of the HTTP endpoints described in this document. HTTP authentication is disabled by default.

mDNS Discovery

Shelly devices announce a HTTP service on port 80 via mDNS. Hostname is always in the form of `shelly<model>-XXXXXXXXXXXX` .

SNTP Time Sync

Shelly devices do not have a built-in real time clock but will automatically synchronize it's time when it is in WiFi Client mode and there is connection to the Internet. Once the time is synchronized devices can execute commands triggered by user-defined weekly schedule or based on sunrise and sunset times. Geolocation data used for sun events is obtained automatically from the public IP address of the device.

Cloud

Shelly devices can report their settings and state to an Internet connected cloud service. The cloud service can modify the settings and change the device state. All communication is over SSL. This service allows device monitoring and control over the Internet using the accompanying mobile applications.

CoAP

Shelly devices broadcast their state in the local network over UDP messages. If you require real time status updates or have other uses for CoAP-based signaling, please ask for additional details.

Based on Mongoose-OS

Shelly devices are built on top of, and along with Mongoose-OS. Mongoose provides an integrated framework for secure sockets, over-the-air updates, application storage, common device housekeeping tasks and more, that are making the reliability and security of the Shelly portfolio possible.

Feedback

If you find issues with this documentation or have other questions or comments about Shelly devices, please email developers@shelly.cloud

Common HTTP API

This section documents the HTTP API implemented by all Shelly devices, which defines their common traits:

- WiFi configuration
- Cloud settings
- HTTP authentication settings
- Firmware updates, identification and other system functions

For every Shelly device, one should consult this section, along with the chapter dedicated to the Shelly

model in question.

HTTP dialect

All properly formed requests return a JSON-encoded payload with an `application/json` MIME type. The meaning of values is described as **Attributes** for each documented resource.

Each resource may also accept a list of **Parameters** which should be supplied either as query-string in the URL or as `application/x-www-form-urlencoded` POST payload.

Error responses carry a 4xx HTTP response code and a `text/plain` response body, usually with an informative message for the type of error which occurred.

All resources except for `/shelly` will require Basic HTTP authentication when it is enabled via `/settings/login`.

The HTTP method used for performing any of the requests below is intentionally ignored. Most endpoints will always return their specific json payload and perform actions if query parameters are specified.

Boolean parameters can be given as `1`, `y`, `Y`, `t`, `T` or case-insensitive `true` for true, any other value will be interpreted as false.

`/shelly`

JavaScript

```
GET /shelly

{
    "type": "SHSW-21",
    "mac": "5ECF7F1632E8",
    "auth": true,
    "fw": "20161223-111304/master@2bc16496",
}
```

Provides basic information about the device. It does not require HTTP authentication, even if authentication is enabled globally. This endpoint can be used in conjunction with mDNS for device discovery and identification. It accepts no parameters.

Attributes

Attribute	Type	Description
`type`	string	Shelly model identifier
`mac`	string	MAC address of the device
`auth`	bool	Whether HTTP requests require authentication
`fw`	string	Current firmware version

`/settings`

JavaScript

```
GET /settings

{
    "device": {
        "type": "SHSW-21",
        "mac": "16324CAABBCC",
    },
    "wifi_ap": {
        "enabled": false,
        "ssid": "shellyMODEL-16324CAABBCC",
        "key": ""
    },
    "wifi_sta": {
        "enabled": true,
        "ssid": "Castle",
        "key": "BigSecretKeyForCastle"
    },
    "login": {
        "enabled": false,
        "unprotected": false,
        "username": "admin",
        "password": "admin"
    },
    "name": "shellyMODEL-16324CAABBCC",
    "fw": "20170427-114337/master@79dbb397",
    "cloud": {
        "enabled": true,
        "connected": true
    },
    "timezone": "",
    "time": ""
}
```

Represents device configuration: all devices support a set of common features which are described here.

Look at the device-specific `/settings` endpoint to see how each device extends it.

Parameters

Parameter	Type	Description
`reset`	bool	Will perform a factory reset of the device

Attributes

Attribute	Type	Description
`device.type`	string	Device model identifier
`device.mac`	string	MAC address of the device in hexadecimal
`wifi_ap`	hash	WiFi access point configuration, see `/settings/ap` for details
`wifi_sta`	hash	WiFi client configuration. See `/settings/sta` for details
`login`	hash	credentials used for HTTP Basic authentication for the REST interface. If `enabled` is `true` clients must include an `Authorization: Basic ...` HTTP header with valid credentials when performing TP requests.
`name`	string	unique name of the device.
`fw`	string	current FW version
`cloud.enabled`	bool	cloud enabled flag
`time`	string	current time in HH:MM format if synced

`/settings/ap`

JavaScript

```
GET /settings/ap

{
    "enabled": false,
    "ssid": "shellyswitch-163248",
    "key": ""
}
```

Provides information about the current WiFi AP configuration and allows changes. The returned document is identical to the one returned by `/settings` in the `wifi_ap` key. Shelly devices do not allow the

SSID for AP WiFi mode to be changed.

Parameters are applied immediately. Setting the `enabled` flag for AP mode to `1` will automatically disable STA mode.

Parameters

Parameter	Type	Description
`enabled`	bool	Set to `1` to return the device to AP WiFi mode
`key`	string	WiFi password required for association with the device's AP

Attributes

Attribute	Type	Description
`enabled`	bool	whether AP mode is active.
`ssid`	string	SSID created by the device's AP
`key`	string	WiFi password required for association with the device's AP

`/settings/sta`

JavaScript

```
GET /settings/sta

{
    "enabled": true,
    "ssid": "Castle",
    "key": "BigSecretKeyForCastle"
}
```

Provides information about the current WiFi Client mode configuration and allows changes. The returned document is identical to the one returned by `/settings` in the `wifi_sta` key.

Parameters are applied immediately. Setting the `enabled` flag for STA mode to `1` will automatically disable AP mode.

Parameters

Parameter	Type	Description
`enabled`	bool	Set to `1` to make STA the current WiFi mode
`ssid`	string	The WiFi SSID to associate with
`key`	string	The password required for associating to the given WiFi SSID

Attributes

Attribute	Type	Description
`enabled`	bool	whether STA mode is active.
`ssid`	string	SSID of STA the device will associate with
`key`	string	WiFi password for the selected SSID

`/settings/login`

JavaScript

```
GET /settings/login

{
    "enabled": false,
    "unprotected": false,
    "username": "admin",
    "password": "admin"
}
```

JavaScript

```
GET /settings/login?enabled=1&username=boss&password=thebigone

{
    "enabled": true,
    "unprotected": false,
    "username": "boss",
    "password": "thebigone"
}
```

HTTP authentication configuration: enabled flag, credentials. `unprotected` is initially false and is used by the user interface to show a warning when auth is disabled. If the user wants to keep using Shelly without a password, they can set `unprotected` to hide the warning.

Parameters

Parameter	Type	Description
`username`	string	length between 1 and 50
`password`	string	length between 1 and 50
`enabled`	bool	whether to require HTTP authentication
`unprotected`	bool	whether the user is aware of the risks

Attributes

Attributes are identical with the parameters and their semantics.

`/settings/cloud`

```JavaScript
GET /settings/cloud?enabled=1

{
    "enabled": true
}
```

Can set the “connect to cloud” flag. When set, Shelly will keep a secure connection to Allterco’s servers and allow monitoring and control from anywhere.

`/status`

```
GET /status

{
    "wifi_sta": {
        "connected": true,
        "ssid": "Castle",
        "ip": "192.168.2.65"
    },
    "cloud": {
        "enabled": false,
        "connected": false
    },
    "time": "17:42",
    "has_update": true,
    "ram_total": 50648,
    "ram_free": 38376,
    "uptime": 39
}
```

JavaScript

Encapsulates current device status information. While settings can generally be modified and don't react to the environment, this endpoint provides information about transient data which may change due to external conditions.

Parameters

Parameter	Type	Description
`wifi_sta`	hash	Current status of the WiFi connection
`wifi_sta.ip`	string	IP address assigned to this device by the WiFi router
`cloud`	hash	current cloud connection status
`time`	string	The current hour and minutes, in `HH:MM` format
`has_update`	bool	If a newer firmware version is available
`ram_total`, `ram_free`	number	Total and available amount of system memory in bytes
`uptime`	number	seconds elapsed since boot

`/reboot`

```
GET /reboot

{}
```

JavaScript

When requested will cause a reboot of the device.

Shelly Switch

Shelly Switch is a WiFi-enabled dual-channel relay with a common power meter. To find out more and download the User Manual, visit the product page

https://shelly.cloud/shelly2/

Shelly Switch can operate in two distinct device modes: **Relay** and **Roller Shutter**. Devices are initially programmed to work in **Relay** Mode. The operating mode of Shelly Switch can be set via the `/settings` endpoint. Commands to perform actions can come from:

- A physical input switch (S1/S2)
- HTTP request, trough the local web interface
- A command sent via the cloud
- A weekly-schedule event or a sunrise/sunset-generated event

Factory Reset

If the web interface of the device cannot be accessed, settings can be brought back to default by switching ON and OFF 5 times the physical switch connected to the device, within the first minute after a reboot or power-on.

Relay Mode

In **Relay** mode, each of the two channels is controlled individually and supports the **ON** and **OFF** commands. In this mode, an overpower threshold can be enabled for each of the two channels, via the `max_power` parameter in `/settings`. If set to a value different than 0, the relay will automatically turn off if current power consumption exceeds:

- the value of `max_power` when only one channel is **ON**
- twice the value of `max_power` when both channels are **ON**

Relay channels also support `auto_on` and `auto_off` settings – these are timers in seconds which will turn ON or OFF the channel when it has been turned OFF or ON respectively, from either a physical switch or network command. Thus, the user can set a limit for how long the channel can be **ON** or **OFF**.

Upon power-on, outputs are initialized using one of 4 available strategies:

- Keep output Off
- Turn output On
- Restore the state of the output from before the power loss
- Read the physical switch state and configure the output accordingly.

Physical input switches can be one of:

- a momentary, push-button switch
- a toggle switch with two stable states
- an “edge” switch. In this input mode, every switch state transition causes a toggle of the output state. This can be used with input from existing two-way switch installations.

Note: Setting output state based on inputs on power on is not supported when inputs are configured as momentary or edge.

To control Shelly Switch in **Relay** mode, use these resources:

- `/settings/relay/{index}` to configure the behavior of each channel
- `/relay/{index}` to control and monitor the channel

Roller Mode

In **Roller** mode the device can be used to control bi-directional motor, with optional obstacle detection and safety switch features. Commands are: **Open**, **Close** and **Stop**.

Shelly Switch has a single logical **Roller**, but we index the HTTP resources to allow for future devices with more output channels and **Roller**-s.

When Shelly Switch is in **Roller** mode, use:

- `/settings/roller/{index}` to configure the behavior of this **Roller** controller, including the advanced features described below
- `/roller/{index}` to query the state and send commands

Obstacle detection

In **Roller** mode, the integrated power meter of Shelly Switch can be used to detect motor overload which usually means something obstructing the movement of the door or roller shutter. A set of parameters define the behavior of Shelly Switch when such event occurs. These are:

- the direction of motion for which obstacle protection is enabled
- the action to perform when the event occurs
- number of seconds to wait before triggering the condition – motors draw big initial current which may

be falsely interpreted as an obstacle.

Safety-switch input

While in **Roller** Mode and when inputs are configured as “one button”, the second physical input can be used as a safety input – it can be tied to an end-stop switch or user safety stop button. Its configuration parameters are:

- when to honor events on the safety input: during either direction or any direction of motion
- what to do when the switch is engaged:
  - stop the motion
  - pause the motion, so when the switch is disengaged it continues in the same direction
- which external commands are allowed while the safety inputs is engaged

Shelly Switch: `/settings`

JavaScript

```
GET /settings?mode=relay

{
    "name": "shellyswitch-163248",
    "mode": "relay",
    "max_power": 1840,
    "relays": [ ... ],
    "rollers": [ ... ],
    "meters": [{"power": 0, "is_valid": true}]
}
```

Shelly Switch extends the common `/settings` endpoint with relay and roller-mode specific data, power consumption status and contains the settings for this device’s **Relays** and **Rollers**. Both are indexed, to allow implementation of the same interface on devices with more output channels.

and also exposes `/settings/relay/N` and `/settings/roller/M` for setting the parameters.

Parameters

Parameter	Type	Description
`mode`	string	Accepted values are `relay` and `roller`
`max_power`	number	Overpower threshold in Watts

Attributes

Attribute	Type	Description
`relays`	array of hashes	See `/settings/relay` for explanation of values
`rollers`	array of hashes	See `/settings/roller` for explanation of values
`meters`	array of hashes	Contains the status of the integrated power meter.

meters attributes

Attribute	Type	Description
`is_valid`	boolean	Whether power metering self-checks OK
`power`	number	Current real AC power being drawn, in Watts

Shelly Switch: `/settings/relay/{index}`

JavaScript

```
GET /settings/relay/0

{
    "ison": false,
    "has_timer": false,
    "overpower": false,
    "default_state": "off",
    "btn_type": "toggle",
    "auto_on": 0,
    "auto_off": 0,
    "schedule": true,
    "schedule_on": "XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "schedule_off": "XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "sun": true,
    "sun_on_times": "00000000000000000000000000000",
    "sun_off_times": "00000000000000000000000000000"
}
```

The returned document here is identical to the data returned in `/settings` for the particular output channel in the `relays` array. The attributes match the set of accepted parameters.

Parameters

Parameter	Type	Description
`reset`	any	Submitting a non-empty value will reset settings for this relay output channel to factory defaults.
`default_state`	string	Accepted values: `off`, `on`, `last`, `switch`
`btn_type`	string	Accepted values: `momentary`, `toggle`, `edge`
`auto_on`	number	Automatic flip back timer, seconds. Will engage after turning the channel OFF.
`auto_off`	number	Automatic flip back timer, seconds. Will engage after turning the channel ON.

Shelly Switch: `/settings/roller/{index}`

JavaScript

```
GET /settings/roller/0

{
    "maxtime": 20,
    "default_state": "stop",
    "swap": false,
    "input_mode": "onebutton",
    "button_type": "toggle",
    "state": "stop",
    "power": 0,
    "is_valid": true,
    "safety_switch": false,
    "obstacle_mode": "disabled",
    "obstacle_action": "stop",
    "obstacle_power": 200,
    "obstacle_delay": 1,
    "safety_mode": "while_opening",
    "safety_action": "stop",
    "safety_allowed_on_trigger": "none"
}
```

The returned document here is identical to the data returned in `/settings` as a hash in the `rollers`. The attributes match the set of accepted parameters.

Parameters

Parameter	Type	Description
`reset`	any	Submitting a non-empty value will reset **Roller** settings to factory defaults
`maxtime`	number	The maximum time needed for the mechanism to completely open or close, seconds.
`default_state`	string	One of `stop`, `open`, `close`. This parameters determines the behavior on power-on.
`swap`	boolean	Whether to swap Open and Close directions
`input_mode`	string	One of `openclose` – each input controls one direction of motion, `onebutton` – a single button press cycles through states: open, stop, close, stop …
`btn_type`	string	One of `momentary` or `toggle`.
`obstacle_mode`	string	One of `disabled`, `while_opening`, `while_closing`, `while_moving`.
`obstacle_action`	string	One of `stop`, `reverse`.
`obstacle_power`	number	Power threshold for triggering “obstacle” condition, Watts.
`obstacle_delay`	number	Number of seconds to wait after powering on the motor before obstacle detection is activated
`safety_mode`	string	One of `disabled`, `while_opening`, `while_closing`, `while_moving`
`safety_action`	string	One of `stop`, `pause`, `reverse`.
`safety_allowed_on_trigger`	string	Which commands are allowed while the safety switch is triggered: `none`, `open`, `close`, `all`.

Shelly Switch: `/status`

```
GET /status

"relays": [
{
    "ison": false,
    "has_timer": false,
    "overpower": false,
    "is_valid": true
},
{
    "ison": false,
    "has_timer": false,
    "overpower": false,
    "is_valid": true
}
],
"rollers": [
    {
    "state": "stop",
    "power": 0,
    "is_valid": true,
    "safety_switch": false,
    "stop_reason": "normal",
    "last_direction": "stop"
    }
],
"meters": [
    {
        "power": 0,
        "is_valid": true,
    }
],
```

JavaScript

Shelly Switch adds information about the current state of the output channels, the logical “roller” device and power metering.

Attribute	Type	Description
`relays`	array of hashes	Contains the current state of the relay output channels. See `/relay/N` for description of attributes
`rollers`	array of hashes	Contains the current state of the logical “roller” device. See `/roller/N` for description of attributes
`meters`	array of hashes	Current status of the power meters

Shelly Switch: `/relay/{index}`

Shows current status of each output channel and accepts commands for controlling the channel. This is usable only when device mode is set to `relay` via `/settings` .

Parameters

Parameter	Type	Description
`turn`	string	Accepted values are `on` and `off` . This will turn ON/OFF the respective output channel when request is sent .
`timer`	number	A one-shot flip-back timer in seconds.

Attributes

Attribute	Type	Description
`ison`	boolean	Whether the channel is turned ON or OFF
`has_timer`	boolean	Whether a timer is currently armed for this channel
`overpower`	boolean	Whether an overpower condition turned the channel OFF
`is_valid`	boolean	Whether the associated power meter is functioning correctly

Shelly Switch: `/roller/{index}`

Controls the logical “roller” device and retrieves its current status.

Parameters

Parameter	Type	Description
`go`	string	Accepted values are `open` , `close` and `stop`
`duration`	number	If specified, the motor will move for this period in seconds. If missing, the value of `maxtime` in `/settings/roller/N` will be used.

Attributes

Attribute	Type	Description
`state`	bool	One of `stop`, `open`, `close`
`power`	number	Current power consumption in Watts
`is_valid`	bool	If the power meter functions properly
`safety_switch`	bool	Whether the safety input is currently triggered
`stop_reason`	bool	Last cause for stopping: `normal`, `safety_switch`, `obstacle`
`last_direction`	bool	Last direction of motion, `open` or `close`

Shelly Plug

Shelly Plug is the most intelligent Wi-Fi power outlet on the market today. Visit the product page for a detailed features overview and User Manual:

https://shelly.cloud/shelly-plug/

The operating mode of Shelly Plug can be set via the `/settings` endpoint. Commands to perform actions can come from:

- A physical button
- HTTP request, trough the local web interface
- A command sent via the cloud
- A weekly-schedule event or a sunrise/sunset-generated event

Factory Reset

If the web interface of the device cannot be accessed, settings can be brought back to default by holding the button for more then 10 seconds or until the red LED starts flashing rapidly.

Operating Mode

Shelly Plug supports the **ON** and **OFF** commands. Overpower threshold can be enabled via the `max_power` parameter in `/settings`. If set to a value different than 0, the relay will automatically turn off if current power consumption exceeds the value of `max_power`.

Shelly Plug also support `auto_on` and `auto_off` settings – these are timers in seconds which will turn ON or OFF the plug when it has been turned OFF or ON respectively, from either a physical button or network command. Thus, the user can set a limit for how long the channel can be **ON** or **OFF**.

Upon power-on, outputs are initialized using one of 4 available strategies:

- Keep output Off
- Turn output On
- Restore the state of the output from before the power loss
- Read the physical switch state and configure the output accordingly.

To control Shelly Plug, use these resources:

- `/settings/relay/0` to configure the behavior
- `/relay/0` to control and monitor

Shelly Plug: `/settings`

```JavaScript
GET /settings

{
    "name": "shellyplug-163248",
    "max_power": 3500,
    "relays": [ ... ],
    "meters": [{"power": 0, "is_valid": true}]
}
```

Shelly Plug adds `max_power` to the list of parameters which can be set via the common `/settings` endpoint:

Parameters

Parameter	Type	Description
`max_power`	number	Overpower threshold in Watts

Attributes

Attribute	Type	Description
`relays`	array of hashes	See `/settings/relay` for explanation of values
`meters`	array of hashes	Contains the status of the integrated power meter.
`max_power`	number	Overpower threshold in Watts

meters attributes

Attribute	Type	Description
`is_valid`	boolean	Whether power metering self-checks OK
`power`	number	Current real AC power being drawn, in Watts

Shelly Plug: `/status`

```JavaScript
GET /status

{
    "relays": [{
        "ison": true,
        "has_timer": false,
        "overpower": false}],
    "meters": [{
        "power": 0.01,
        "is_valid": true}]}
    ]
}
```

Shelly Plug adds information about the current state of the output channel (ON or OFF) and instantaneous power reading in Watts.

Attribute	Type	Description
`relays`	array of hashes	Contains the current state of the relay output channels. See `/relay/0` for description of attributes
`meters`	array of hashes	Current status of the power meter, see

Shelly Plug: `/settings/relay/0`

```
GET /settings/relay/0

{
    "ison": false,
    "has_timer": false,
    "overpower": false,
    "default_state": "off",
    "btn_type": "toggle",
    "auto_on": 0,
    "auto_off": 0
}
```

The returned document here is identical to the data returned in `/settings` for the only output channel in the `relays` array. The channel index exists to preserve API compatibility with multi-channel Shelly devices. Attributes in the response match the set of accepted parameters.

Parameters

Parameter	Type	Description
`reset`	any	Submitting a non-empty value will reset settings for this relay output channel to factory defaults.
`default_state`	string	Accepted values: `off`, `on`, `last`, `switch`
`btn_type`	string	Accepted values: `momentary`, `toggle`, `edge`
`auto_on`	number	Automatic flip back timer, seconds. Will engage after turning the channel OFF.
`auto_off`	number	Automatic flip back timer, seconds. Will engage after turning the channel ON.

Shelly Plug: `/relay/0`

Shows current status of the output channel and accepts commands for controlling it.

Parameters

Parameter	Type	Description
`turn`	string	Accepted values are `on` and `off`. This will turn ON/OFF the respective output channel when request is sent .
`timer`	number	A one-shot flip-back timer in seconds.

Attributes

Attribute	Type	Description
`ison`	boolean	Whether the channel is turned ON or OFF
`has_timer`	boolean	Whether a timer is currently armed for this channel
`overpower`	boolean	Whether an overpower condition turned the channel OFF