# Example prototype for SWOTY

Documentation of a REST API for a thing which is able to communicate with SWOTY network.

## API

### API - Activate a specific function

Example route representing a fictional function of this thing. This url may mostly be arbitary, but it must match one of the URLs provided in the /functions response. SWOTY posts to this route if the user wants to active this function. At least the write token should be present.

The response of this route is not important to SWOTY. If this things wants to notify SWOTY about changes like starting to do something or finished doing something, it may post these information in form of status updates to SWOTY (see SWOTY REST API reference).

POST

```
/functions/activateexamplefunction
```

- Example usage: [#examples-API-activate_a_function-0_0_0-0]

```
curl --header "accesstoken: 12345" -i http://localhost:3000/functions/activateexamplefunction
```

#### Header

| Feld | Typ | Beschreibung |
|------|-----|--------------|
| accesstoken | String | Access token to activate this function. Should at least be the write token. |

- Header-Example: [#header-examples-API-activate_a_function-0_0_0-0]

```
{
   "accesstoken": "12345"
}
```

#### Success 200

| Feld | Typ | Beschreibung |
|------|-----|--------------|
| message                    optional | String | Success message. |

- Success-Response: [#success-examples-API-activate_a_function-0_0_0-0]

```
HTTP/1.1 200 OK
{
   "message" : "Function activated."
}
```

### API - Deregister this thing

Deregister Action. SWOTY calls this route, when a user wants to remove this thing from his profile. This may be in case he/she doesn't want this thing anymore or he/she wants to give it to another user. The internal access token/register token should be marked as unused so that device to ready to be registered again.

The owner token should be passed to activate this function. The token is passed by request header "accesstoken".

GET

```
/deregister
```

- Example usage: [#examples-API-deregister-0_0_0-0]

```
curl --header "accesstoken: 12345" -i http://localhost:3000/deregister
```

## Header

| Feld | Typ | Beschreibung |
|------|-----|--------------|
| accesstoken | String | Access token to activate this function. Should be the owner token. |

- Header-Example: [#header-examples-API-deregister-0_0_0-0]

```
{
   "accesstoken": "12345"
}
```

## Success 200

| Feld | | Typ | Beschreibung |
|------|--|-----|--------------|
| message | optional | String | Success message. |

- Success-Response: [#success-examples-API-deregister-0_0_0-0]

```
HTTP/1.1 200 OK
{
   "message" : "Device deregistered."
}
```

# API - Information about this thing and it's status.

Returns a list of information about this thing. These information should represent the current status. If the status changes, SWOTY should be notified about that. To do that, just make a POST request to SWOTY, the network will then retrieve the updated information from this route, See SWOTY REST Api documentation for more details.

GET

```
/functions/information
```

- Example usage: [#examples-API-information-0_0_0-0]

```
curl --header "accesstoken: 12345" -i http://localhost:3000/information
```

## Header

| Feld | Typ | Beschreibung |
|------|-----|--------------|
| accesstoken | String | Access token used to retrieve the thing's information. Should at be the read token. |

- Header-Example: [#header-examples-API-information-0_0_0-0]

```
{
   "accesstoken": "12345"
}
```

## key-value type

| Feld | Typ | Beschreibung |
|------|-----|--------------|
| information | Object[] | Example for key-value type. See example response for key-value for sample data. |
| title | String | Set title to "Status". |

| | | String | Omitted in this example to force a key value type. |
|---|---|---|---|
| type | optional | | |
| value | | String | Set raw value to "working". |

## boolean type

| Feld | Typ | Beschreibung |
|---|---|---|
| information | Object[] | Example for boolean type. See example response for boolean for sample data. |
| title | String | Set title to "Working". |
| type | String | Set type to boolean. Swoty will render a green/red badge with the label "yes"/"no" for true/false values. |
| value | String | Sets value to true so SWOTY will render a green badge with the label "yes". |

## table type

| Feld | Typ | Beschreibung |
|---|---|---|
| information | Object[] | Example for table type. See example response for table for sample data. |
| title | String | Here we use aa bus station as an example which displays a table with bus lines and their arrival times. |
| type | String | Use type "table" so SWOTY will display the data as a HTML table. |
| value | Object | When using the table type, the value is not a primitive anymore but a complex JSON object containing the table headers and rows. |
| header | String[] | An Array containing the html table headers. Here we will use "Bus line" and "Arrival time" as headers. |
| data | String[][] | The table rows. This is an array containing arrays. Every nested array represents one table row. Here we define to arriving buses, line 11 arriving at 15:22 and line 5 arriving at 16:15. |

## percentage type

| Feld | Typ | Beschreibung |
|---|---|---|
| information | Object[] | Example for percentage type. See example response for percentage for sample data. |
| title | String | Example for a progress bar using percentage type. |
| type | String | Use percentage type by providing "percentage" as type. |
| value | String | Percent finished (progress) from 0 to 100. |

## html type

| Feld | Typ | Beschreibung |
|---|---|---|
| information | Object[] | Example for html type. See example response for html for sample data. |
| title | String | |

| | | Title for the html content. SWOTY will show the title prior to the html formatted content. |
|---|---|---|
| type | String | Use html type by providing "html" as type. |
| value | String | Some static cotent with extra HTML markup. |

## Success 200

| Feld | Typ | Beschreibung |
|---|---|---|
| information | Object[] | List of information about this thing. |
| title | String | The title for the information. SWOTY will use this as title. |
| type                    optional | String | The type of data. Currently supported:<br><br>• boolean<br>• table<br>• percentage<br>• html<br>• if no type is provided, SWOTY will display the raw value (escaped)<br><br>See other example responses for detailed useage |

- (key-value tye) {json} Success-Response with boolean type: [#success-examples-API-information-0_0_0-0]
- (boolean tye) {json} Success-Response with boolean type: [#success-examples-API-information-0_0_0-1]
- (table tye) {json} Success-Response with boolean type: [#success-examples-API-information-0_0_0-2]
- (percentage tye) {json} Success-Response with percentage type: [#success-examples-API-information-0_0_0-3]
- (html tye) {json} Success-Response with html type: [#success-examples-API-information-0_0_0-4]
- Success-Response: [#success-examples-API-information-0_0_0-5]

```
HTTP/1.1 200 OK
{
    "information": [
      {
          "title" : "Status",
          "value" : "working"
      }
    ]
  }
```

```
HTTP/1.1 200 OK
{
    "information": [
      {
          "title": "Working",
          "type": "boolean",
          "value": true
      }
    ]
  }
```

```
HTTP/1.1 200 OK
{
    "information": [
      {
          "title" : "arrivals",
          "type" : "table",
          "value" : {
              "header" : ["Bus line", "Arrival time"],
              "data" : [
                  ["11", "15:22"],
                  ["5", "16:15"]
              ]
          }
      }
    ]
```

```
        }
```

```
HTTP/1.1 200 OK
{
    "information": [
        {
            "title" : "Progress making coffee",
            "type" : "percentage",
            "value" : "0"
        }
    ]
}
```

```
HTTP/1.1 200 OK
{
    "information": [
        {
            "title" : "value",
            "type" : "html",
            "value" : "<div><b>html value example</b></div>"
        }
    ]
}
```

```
HTTP/1.1 200 OK
{
    "information": [
        {
            "title" : "Status",
            "value" : "working"
        },

        {
            "title": "Working",
            "type": "boolean",
            "value": true
        },

        {
            "title" : "value",
            "type" : "table",
            "value" : {
                "header" : ["header1", "header2", "header3"],
                "data" : [
                    ["value1", "value2", "value3"],
                    ["value1", "value2", "value3"]
                ]
            }
        },

        {
            "title" : "Progress making coffee",
            "type" : "percentage",
            "value" : "0"
        },

        {
            "title" : "value",
            "type" : "html",
            "value" : "<div><b>html value example</b></div>"
        }
    ]
}
```

# API - List of available functions

This route should give a list of the available functions which can be activated by a REST call. Every function contains:

- a unique name for identification
- an URL which is called to activate the function
- an indicator if this function is available

- a list of parameters

Every parameters has

- a unique name
- a type which will determine the representation on SWOTY
- an indicator if the parameter is optional / required
- a list of constraints

Every constraint has:

- a type
- an error message if the user input (on SWOTY) does not match the constraint. This means: every input of the user will be validated server side by SWOTY before submitting the data to the network!

If the functions of this thing change anytime after registrations, SWOTY should be notified about that. (See Example call to Swoty to notify a change in the function list)

GET

```
/functions
```

- Example call to Swoty to notify a change in the function list [#examples-API-functions-0_0_0-0]
- Call by SWOTY to retrieve function list [#examples-API-functions-0_0_0-1]

```
curl -i --header "accesstoken: mynetworkaccestoken" http://urltoswoty.com/api/v1/information/update
```

```
curl -i --header "accesstoken: 12345" http://localhost:3000/functions
```

## Header

| Feld | Typ | Beschreibung |
| --- | --- | --- |
| accesstoken | String | Access token to retrieve function list. Should be the read token. |

- Header-Example: [#header-examples-API-functions-0_0_0-0]

```
{
  "accesstoken": "12345"
}
```

## Success 200

| Feld | Typ | Beschreibung |
| --- | --- | --- |
| functions | Object[] | List of available functions |
| name | String | Unique name of the function |
| url | String | Url used by SWOTY to activate this function (POST) |
| available | String | Indicates if this function may be called at the moment |
| parameters | Object[] | A List of parameters required to activate this function. Every parameter will be set by user input. Representation depends on the parameter type. |
| name | String | The name of the parameter. Will be used as the label text for the corresponding input. Should be unique withing the function. Other functions may vontain a parameter with the same name. |
| type | String | The type of the parameter. Currently supported: <ul><li>integer</li></ul> |

- Choice
- text

| choices | optional | String[] | Required when using "Choice" as parameter type. Contains a list of predefined values / options. SWOTY will send the index of the selected choice with an activate function request. |
|---|---|---|---|
| multiple | optional | Boolean | Only evaluated when using "Choice" as parameter type. Lets the user select multiple values from the choice list.<br><br>Standardwert: `false` |
| expanded | optional | Boolean | Only evaluated when using "Choice" as parameter type.<br><br>- false: representation will be a dropdown list<br>- true: representation will be a radio select list Note: the represenattion will vary if multiple is set to true<br><br>Standardwert: `false` |
| required | | Boolean | Indicates if the user must fill in this value. |
| readOnly | optional | Boolean | Indicates if the parameter value is editable. If true, only the default value will be displayed if provided.<br><br>Standardwert: `false` |
| defaultValue | optional | String | The default value for this parameter. |
| constraints | optional | Object[] | A list of contraints for this parameter. SWOTY will validate the form input server side by this list before sending it to this thing. |
| type | | String | The type of this constraint. Currently supperted:<br><br>- NotNull<br>- NotBlank<br>- Date<br>- DateTime<br>- Choice<br>- Country<br>- GreatedThan<br>- LessThan<br>- Language<br>- Locale<br>- Time |
| message | | String | This message will be shown to the user if his input does not validate. |

- Example Success-Response (tries to list all available configuration options): [#success-examples-API-functions-0_0_0-0]

```
HTTP/1.1 200 OK
{
    "functions": [{
            "name": "Function-A",
            "url": "http://www.example.com",
            "available": true,
            "parameters": [
                {
                    "name": "Param-A",
                    "type": "integer",
                    "required": true,
                    "constraints": [
                        {
                            "type": "NotNull",
                            "message": "Param A may not be null"
```

```json
                    }
                ]
            },
            {
                "name": "Param-B",
                "type": "Choice",
                "choices": [
                    1, 2, 3, 4, 5
                ],
                "required": false,
                "constraints": [
                    {
                        "type": "NotBlank",
                        "message": "Param B may not be blank"
                    }
                ]
            },
            {
                "name": "Param-C",
                "type": "Choice",
                "multiple": true,
                "expanded": false,
                "choices": [
                    1, 2, 3, 4, 5
                ],
                "required": false,
                "constraints": [
                    {
                        "type": "NotBlank",
                        "message": "Param C may not be blank"
                    }
                ]
            },
            {
                "name": "Param-D",
                "type": "Choice",
                "multiple": true,
                "expanded": true,
                "choices": [
                    1, 2, 3, 4, 5
                ],
                "required": false,
                "constraints": [
                    {
                        "type": "NotBlank",
                        "message": "Param D may not be blank"
                    }
                ]
            }
        ]}, {
            "name": "Function-B",
            "url": "http://www.example.com",
            "available": true,
            "parameters": [
                {
                    "name": "Param-2A",
                    "type": "text",
                    "required": true,
                    "constraints": [
                        {
                            "type": "Locale",
                            "message": "Param 2A must be a locale"
                        }
                    ],
                    "defaultValue": "hallo"
                },
                {
                    "name": "Param-2B",
                    "type": "integer",
                    "required": false,
                    "readOnly": true,
                    "defaultValue": 20
                }
            ]
        }]
    }
```

# API - Register this thing in SWOTY network

The route is used to register this thing in SWOTY. The URL .../register is just convention, the real URL is provided by the QR Code of this thing (which is scanned by the user / SWOTY). The the manufacturer of this thing is responsible himself that the URL on the QR code matches this route.

The QR code should also provide a register token which will be sent with the register request. The register token acts like a serial key. It asserts, that the device may only be registered by the user who owns the thing (and thus the QR code). A copy of the register token should be placed on software of this thing (maybe hardcoded on the filesystem) to check if the provided token is correct.

Further more, this thing should mark the register token as used after the device has been registered so that no more people can register this thing. This token will uniquely identify this thing and authorize it for posting messages to SWOTY.

SWOTY will append a so called network access token to the URL as a parameter. This token should be saved on the thing and sent with every request to SWOTY.

As response, this route should provide SWOTY all relevant information needed to communicate with this thing. See example success response for more details.

GET

```
/register
```

- Example usage: [#examples-API-register-0_0_0-0]

```
curl -i http://localhost:3000/register?access_token=12345&networktoken=13579
```

## Parameter

| Feld | Typ | Beschreibung |
|---|---|---|
| access_token | String | Accesstoken or register token. This is the included token in the QR code. Once used, this token should not work anymore. |
| networktoken | String | Accesstoken for this thing giving it permissions to POST to SWOTY. |

## Success 200

| Feld | Typ | Beschreibung |
|---|---|---|
| device | Object | Device information. |
| id | String | Internal id of the device. |
| classification | String | Classification of this thing. Currently not used by SWOTY. |
| url | String | Base (absolute) URL to the thing's api endpoint. All URLs like deregister, functions and information will be generated by SWOTY by convention. |
| profileimage optional | String | Optional image repreenting this thing. If no image is provided, SWOTY will use a default image. |
| tokens | Object | Object defining all available access tokens SWOTY will use to access this thing. |
| owner_token | String | Owner token. Used for administrative actions like deregister. |
| write_token | String | Write token. Used to activate functions. |
| read_token | String | Read token. Used to read information about this thing. |

- Success-Response: [#success-examples-API-register-0_0_0-0]

```
HTTP/1.1 200 OK
{
  "device": {
      "id": 12345,
      "classification": "sdf",
      "url": "http://.....",
      "profileimage": "http://....",
      "tokens": {
          "owner_token": "293487239",
          "write_token": "2305982304",
          "read_token": "23405834905"
      }
  }
}
```

Erstellt mit apidoc [http://apidocjs.com] 0.12.3 - 2015-04-26T14:58:59.496Z