Запросы типа Message и Processed отправляются контроллером всегда в разных запросах.

В одном запросе всегда одно сообщение. Одно сообщение содержит массив данных для множества ламп.

# Index.php

Из потока POST читается json-строка и сохраняется в mesginputh как есть.

Полученная строка преобразуется в объект типа Dictionary: ключ-значение.

Проверяется тип сообщения:

## Processed

{

"ControllerID": "9",

"Processed": {

"51": {

"status": "Error",

"val": "Not online"

},

"52": {

"status": "Error",

"val": "Not online"

}

}

Из объекта выделяются MessageID (51,52) и объект с данными status и val.

Для тех сообщений, где "status" == "OK", в таблице message в поле Status записывается 2.

Для тех сообщений, где "status" == "Error", в таблице message в поле Status записывается 0, в поле Answer записывается значение из поля val объекта с данными   
(например: "Not online").

## Message

{

"Message": {

"807B8590200006DE": {

"Tmp": null,

"LastDateTime": {

"Date": "25.05.2019",

"Time": "20:27:49"

},

"L": null,

"ArcPwr": "100",

"State": "1",

"W": null,

"DT": null,

"GroupID": "0",

"IsAlive": false,

"P": "100",

"TimeIsOut": true,

"SN": "807B8590200006DE",

"RSSI": "-84"

},

"36FF46550DA7A6CA": {

"Tmp": null,

"LastDateTime": {

"Date": null

},

"L": null,

"ArcPwr": "100",

"State": "1",

"W": null,

"DT": null,

"GroupID": "0",

"IsAlive": false,

"P": "100",

"TimeIsOut": true,

"SN": "36FF46550DA7A6CA",

"RSSI": null

},

. . .

"ControllerID": "7"

}

Ключ объекта с данными содержит серийный номер лампы, тоже самое, что и поле SN объекта с данными.

Для каждого объекта с данными по серийному номеру лампы проверяется наличие связанных записей в таблицах setting и condition.

Если нет записи в setting, то полученные данные заносятся только в таблицу msginput. На этом обработка прекращается.  
(*Получается, что эта таблица не лишняя, а служит для сохранения данных о лампах, которые ещё не прописаны в базе должным образом*)

Если нет записи в condition, то запись создаётся.

Проверяются разные поля типа State, IsAlive, TimeIsOut и др., вычисляется значение для поля Status, переводится в двоичный формат; определяется группа etc.

Обновляется запись в таблице condition полученными данными.

В таблицу conditionh добавляется запись с полученными данными.

## Очередь команд/расписания

Перед окончанием обработки запроса от контроллера, проверяется очередь сообщений в таблице message. Новые сообщения со статусом “0” или старые сообщения, которые не были обработаны, собираются в json-объект и возвращаются контроллеру.

{

"552": {

"method":"POST",

"val":"GOOFF",

"path":"lamps/fd00::212:4b00:17e3:e7ce/Cmd"

}

"553": {

"method":"POST",

"val":"GOOFF",

"path":"lamps/fd00::212:4b00:17e3:e7ce/Cmd"

}

}

Для этих сообщений в таблице message поле Status устанавливается в “1”.

# Web Api

Для каждого типа сообщений используется свой адрес. Для каждой версии используется свой префикс в адресе ресурса.

Например:

[http://light.sofiot.com/**api**/message/](http://light.sofiot.com/api/message/)

[http://light.sofiot.com/**api-ver0012**/message/](http://light.sofiot.com/api-ver0012/message/)

[http://light.sofiot.com/**api**/processed/](http://light.sofiot.com/api/processed/)

[http://light.sofiot.com/**api-ver0011**/processed/](http://light.sofiot.com/api-ver0011/processed/)

Каждый обработчик запросов контроллера (граничного раутера) сохраняет текст запроса в виде json-строки в таблицу mesginputh (как и index.php).

Алгоритмы схожи, отличаются только форматы входящих и возвращаемых данных.

## Processed

Формат немного отличается. Вместо объекта типа Dictionary, используется массив объектов:

{

"ControllerID": "9",

"Processed": [

{

"MessageID": "511",

"Result": {

"status": "Error",

"val": "lamps/fd00::212:4b00:1680:7c62/Cmd not found"

}

},

{

"MessageID": "512",

"Result": {

"status": "Error",

"val": "lamps/fd00::212:4b00:1680:7c62/Cmd not found"

}

}

]

}

## Message

{

"ControllerID": "9",

"Message": [

{

"SN": "fd00::212:4b00:1bca:72a3",

"State": "80",

"LastDateTime": "2020-04-07T13:46:49",

"ArcPwr": "100",

"Tmp": 76,

"L": 500,

"P": "100",

"W": null,

"DT": null,

"RSSI": 122,

"TimeIsOut": false,

"IsAlive": true,

"GroupID": "0"

},

{

"SN": "fd00::212:4b00:1bca:72a3",

"State": "E4",

"LastDateTime": "2020-04-07T13:46:49",

"ArcPwr": "100",

"Tmp": 76,

"L": 500,

"P": "100",

"W": null,

"DT": null,

"RSSI": 122,

"TimeIsOut": false,

"IsAlive": true,

"GroupID": "0"

},

{

"SN": "fd00::212:4b00:1bca:7290",

"State": "B2",

"LastDateTime": "2020-04-07T13:46:49",

"ArcPwr": "50",

"Tmp": 76,

"L": 555,

"P": "50",

"W": null,

"DT": null,

"RSSI": -88,

"TimeIsOut": false,

"IsAlive": true,

"GroupID": "0"

}

]

}

## Очередь команд/расписания

[{

"messageID" : 5101,

"message" : {

"method" : "POST",

"val" : "GOOFF",

"path" : "lamps/fd00::212:4b00:1680:7c62/Cmd"

}

},

{

"messageID" : 5102,

"message" : {

"method" : "POST",

"val" : "GOON",

"path" : "lamps/fd00::212:4b00:1680:7c62/Cmd"

}

}

},

. . .

]

Или

[{

"messageID" : 5011,

"message" : {

"method" : "POST",

"val" : [

{"C":"GOSC4","T":"20:00"},

{"C":"GOOFF","T":"07:00"}

],

"path" : "lamps/fd00::212:4b00:1680:7c62/schs"

}

},

. . .

]