# API Documentation

## Print API

### API details

1. url: http://print.familyhardware.com/api/2/print/
2. method: POST
3. Request Body in JSON format:

```
{
    "devicemac": "00:11:62:30:41:cd",
    "drawer": {
        "openAt": "START"
    },
    "buzzerCount": {
        "start": 2
    },
    "print_json": [
        {
            "align": "left",
            "type": "IMAGE",
            "url": "https://star-emea.com/wp-
content/uploads/2015/01/logo.jpg",
            "width": "60%",
            "min_width": "48mm"
        },
        {
            "type": "LINE",
            "mode": "UP"
        },
        {
            "align": "centre",
            "type": "CONTENT",
            "content": [
                "",
                "<w3h3><b>Family Hardware</b></w3h3>",
                ""
                ]
        },
        {
            "type": "LINE",
            "mode": "DOWN"
        },
        {
            "align": "centre",
            "type": "CONTENT",
            "content": [
                "FamilyHardware",
                "Address 1",
```

```
                "Address 2",
                "Phone number",
                ""
            ]
        },
        {
            "type": "LINE",
            "mode": "UP"
        },
        {
            "align": "left",
            "type": "COLUMNS",
            "value": {
                "indent": "0mm",
                "columns": [
                    {
                        "left": "SKU",
                        "short": "item 1",
                        "right": "price",
                        "sub":[
                            {
                                "left": "Item Name",
                                "right": "Quantity"
                            }
                        ]
                    }
                ]
            }
        },
        {
            "type": "LINE",
            "mode": "DASHED"
        },
        {
            "align": "left",
            "type": "COLUMNS",
            "value": {
                "indent": "0mm",
                "columns": [
                    {
                        "left": "SKU 1232131",
                        "short": "item 1",
                        "right": "$0.20",
                        "sub":[
                            {
                                "left": "one item",
                                "right": "x 12"
                            }
                        ]
                    },
                    {
                        "left": "SKU 342422",
                        "short": "item 2",
                        "right": "$0.30",
```

```
                        "sub":[
                            {
                                "left": "two item",
                                "right": "x 12"
                            }
                        ]
                    },
                    {
                        "left": "item name 3",
                        "short": "item 2",
                        "right": "$0.40",
                        "sub":[
                            {
                                "left": "third item",
                                "right": "x 12"
                            }
                        ]
                    },
                    {
                        "left": "item name 4",
                        "short": "item 4",
                        "right": "$0.50",
                        "sub":[
                            {
                                "left": "fourth item",
                                "right": "x 12"
                            }
                        ]
                    },
                    {
                        "left": "item name 3",
                        "short": "item 2",
                        "right": "$0.40",
                        "sub":[
                            {
                                "left": "fifth item",
                                "right": "x 12"
                            }
                        ]
                    },
                    {
                        "left": "item name 4",
                        "short": "item 4",
                        "right": "$0.50"
                    }
                ]
            }
        },
        {
            "type": "LINE",
            "mode": "DASHED"
        },
        {
            "align": "right",
```

```
            "type": "CONTENT",
            "content": [
                "",
                "Total Amount:  $30",
                ""
            ]
        }
    ]
}
```

4. Response:

```
{
    "message": null,
    "position": "189"
}
```

## Request fields in details

1. devicemac *string*

   Takes device mac address only. e.g. 00:11:62:30:41:cd

2. Drawer *dict*

    1. Drawer open has 2 modes, at start and at the end.

    2. `openAt` accepts 3 values *`START|END|NONE`*.

   ```
   "drawer": {
       "openAt": "START"
   }
   ```

3. Buzzer counts

    1. It has 2 keys, at the start and at the end.\

    2. The keys take integers as values to determine the number of times it should buzz.

    3. Maximum value allowed is 20.

   ```
   "buzzerCount": {
       "start": 2,
       "end": 3
   }
   ```

4. print_json *list of dictionaries*

List of dictionaries of different item types. We would first define what is "`type`". `type` is a requried field of every dictionary within each list of `print_json` field. `type`s are as followed:

1. `CONTENT`

```
{
    "type": "CONTENT",
    "align": "centre",
    "content": [
        "",
        "<w3h3><b>Family Hardware</b></w3h3>",
        ""
        ]
}
```

We have the `type` field, which has a `CONTENT` value. This dictionary type has valid fields as followed:

1. type: `*CONTENT*`
2. content *list of string. Empty string will be considered as a new line.*
3. align `left|right|centre`

Tags in `type` as *`CONTENT`*:

1. Bold: As an example `<b>Hello World<b>` is a valid value.
2. Text Size: Content can be sized by adjusting the heifht and width of the text elements. e.g. `<w2h2>Hello</w2h2> world` will increase the width of the hello twice the width and height of the standard text of the printout.

**Tags ranging from to are valid tags. Tags are only valid for `type` as `CONTENT`**

2. `IMAGE`

```
{
    "type": "IMAGE",
    "align": "left",
    "url": "https://star-emea.com/wp-
content/uploads/2015/01/logo.jpg",
    "width": "60%",
    "min_width": "48mm"
}
```

The `type` field has `IMAGE` as it's value. This dictionary represents the image content.

This dictionary has following valid fields:

1. type: *IMAGE*

2. align: *left|right|centre*
3. url: *link to any image file*
4. width: *in percentage*
5. min_width: It will overwrite the size of the image if the size calcualated from the image percentage is less than min_width.

3. `COLUMNS`:

```
{
    "type": "COLUMNS",
    "align": "left",
    "value": {
        "indent": "0mm",
        "columns": [
            {
                "left": "SKU",
                "short": "item 1",
                "right": "price",
                "sub":[
                    {
                        "left": "Item Name",
                        "right": "Quantity"
                    }
                ]
            }
        ]
    }
}
```

The `type` field has `COLUMNS` as its value. This dictionary represents the columns content.

Valid fields:

1. type: `COLUMNS`
2. align: `left|right|centre`
3. value: `dict`
    1. indent: Space indent from left to right in mm. e.g. 40mm
    2. columns: `list of rows as dictionary`
        1. left: left column value
        2. right: right column value
        3. short: If the left value has more characters than it could fit in the paper then short would woverride left value and visible in the print.

4. sub: list of subrows
    1. left: left column value of the subrow.
    2. right: right column value of the subrow.

4. `Line`:

```
{
    "type": "LINE",
    "mode": "DASHED"
},
```

The `type` field has `LINE` as it's value. This dictionary represents line in the print outs.

Valid fields:

1. type: `LINE`
2. mode: *`DASHED|UP|DOWN`*

5. `Barcode`:

```
{
    "align": "left",
    "type": "barcode",
    "barcode_type": "code128",
    "data": "34242346",
    "hri": true
}
```

The `type` field has barcode as it's value. This dictionary represents the barcode details that should be printed as the output.

Valid fields:

1. type: *barcode*
2. align: *left*|*right*|*centre*
3. barcode_type: "code128" Following are the valid barcode values: `ean8, jan8, ean13, jan13, upc_e, upc_a, itf, code39, code93, code128, nw7, qr, pdf417`
4. data: e.g. SKU 1238338
5. hri: true|false. If you would like to display the data under the generated barcode, you could set hri as true. hri = human redeable information.

# Label Print API

## API details

1. url: http://print.familyhardware.com/api/2/labelprint/

2. method: POST
3. Request Body in JSON format:

```
{
    "devicemac": "00:11:62:0e:27:69",
    "print_labels": [
        {
            "barcode_type": "code128",
            "hri": true,
            "sku": "SKU 12313232",
            "product_name": "CP 1234567890 LASER GUN ONE TWO",
            "amount": "$1000.20",
            "qty_desc": "each"
        }
    ]
}
```

Request description:

| **propeties** | **required?** | **details** |
|---|---|---|
| devicemac | Yes | Mac id of the label printer e.g. 00:11:62:0e:27:69 |
| print_labels | Yes | List of labels to be printed. Currently it allows only one label print at a time. |
| **print_labels** | | |
| barcode_type | Yes | code128 is a valid type |
| hri | Yes | Boolean, accepts `true` or `false` |
| product_name | Yes | Name of the product |
| amount | Yes | String, accepts values such as `$9999.20` |
| qty_desc | Yes | A short description about the quantity available for the amount |