

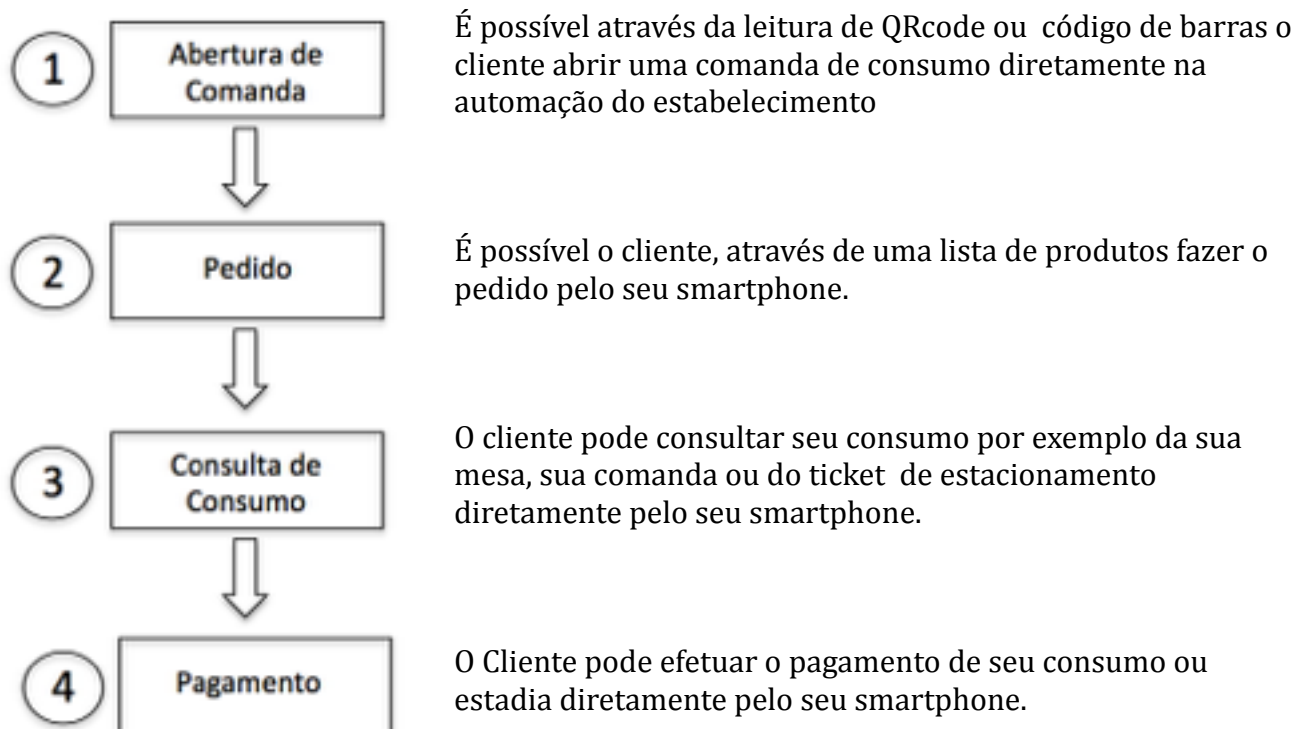


Documentação de integração MO2all com automações  
Vs. 2.1  
01/07/2015

# Documento de integração MOB2all com automação

O MOB2all possibilita que o cliente efetue 4 tipos de operação dependendo do tipo de automação em que o estabelecimento utiliza.

## Operações



Sendo que todas as operações podem ser acessadas e habilitadas pelo estabelecimento de forma independente de acordo com sua necessidade de negócio.

Isto quer dizer que mesmo que a automação permita o acesso ao menu, abertura de comanda o estabelecimento pode optar em somente permitir consulta de consumo e pagamento ou somente o pagamento e assim por diante.

## Abertura de Comanda

Na opção de abertura de comanda o estabelecimento pode optar se uma comanda/mesa poderá ser dividida por mais de um cliente, exemplo, vários clientes abrem uma mesma comanda de uma mesa de restaurante/bar; Ou se a comanda poderá ser aberta somente por um único cliente como exemplo, boates, padarias, etc.

Toda vez que um cliente abre uma comanda, automaticamente é realizado uma consulta de consumo na mesma. Com isso é possível que um cliente possa identificar se a comanda está vazia ou quanto já consumiram no momento que ele entrou na comanda.

## Pedido

O MOB2all pode realizar pedidos diretamente na automação.

Essa opção pode ser configurada de duas formas Loop único ou Loop infinito.

---

## Loop único

Indicado para restaurantes tipo fastfood e delivery, pois o pedido somente é registrado na automação após o pagamento.

Exemplo: O cliente acessa o menu do restaurante, escolhe um prato, bebidas, etc, consulta o subtotal e realiza o pagamento pelo próprio celular. O pedido somente será registrado pela automação após o pagamento.

---

## Loop Infinito

Indicado para restaurante do tipo “a la carte”, pois os pedidos e abertura de comanda podem ser executados até o fechamento da conta.

Uma vez a conta fechada pedidos e abertura de comanda ficam indisponíveis até o pagamento da conta.

## Consulta

O MOB2all permite que o cliente possa consultar seu consumo. Essa consulta pode ser de uma comanda aberta ou consultando um código de barras ou QRCode.

No caso de consulta de uma comanda aberta, o aplicativo mostra os itens pedidos com seus valores e o valor parcial total de consumo.

No caso de um código de barra ou QRCode, ticket de estacionamento ou Valet, é possível consultar o valor a ser pago naquele momento.

## Pagamento

O pagamento é realizado diretamente através da afiliação do estabelecimento com o adquirente e o cartão do cliente cadastrado no aplicativo MOB2all.

Uma vez consultado o valor a ser pago é possível realizar 4 tipos de pagamento de acordo com a funcionalidade da automação.

- Pagamento por Valor
- Pagamento por Produto
- Pagamento por Pessoa
- Pagamento Total

# WEB\_Based

## Informações Técnicas

Toda mensageria segue o padrão JSON

## Retorno

O retorno de todas as requisições respeitam o seguinte formato:

Parâmetro	Descrição	Tipo
action	Ação referente a requisição	String
client	CNPJ	String
message	Mensagem	String
own	ID da Operação - Toda operação tem esse código de identificação	Numérico
response	True ou False	Boolean
type	Socket	String

```

{
  "action": "RESULT",
  "client": "NTK-TESTE",
  "message": {
    },
  "own": 1704,
  "response": true,
  "type": "SOCKET"
}

```

## Acesso ao MENU

Obter o catálogo de produtos por categoria do estabelecimento.

Método GET

[http://url\\_estabelecimento.com.br/automation/menu?type=mob2all](http://url_estabelecimento.com.br/automation/menu?type=mob2all)

Parâmetro	Descrição
menu	Tipo do serviço
type	Tipo de menu

## Retorno

Parâmetro	Descrição	Tipo
nameCat	Nome da categoria	String
idCat	identificação da categoria	INT
url_image	Url para exibição de imagens	String
itens	lista de itens	Array
name	nome do item	String
idItem	identificação do item	INT
cost	custo do item	Float
amount	quantidade	Float
order	ordem de views	INT
obs	observações	Array

```

"data": [
  {
    "nameCat": "SUCOS",
    "idCat": 6,
    "url_image": "https://mob.2all.com.br/imagem...",
    "itens": [
      {
        "name": "Iogurte",
        "idItem": 75,
        "url_image": "https://mob.2all.com.br/imagem...",
        "cost": 0.000000,
        "amount": 1.3,
        "order": 1,
      },
      {
        "name": "GUARANA",
        "idItem": 3,
        "url_image": "https://mob.2all.com.br/imagem...",
        "cost": 2.000000,
        "amount": 0,
        "order": 2,
        "obs": [
          "Com Gelo"
        ]
      }
    ]
  }
]

```

## Abertura de Comanda e fast food e consulta

### Pedidos

Obter itens adquiridos.

Método GET

[http://url\\_estabelecimento.com.br/consumption?code={numerico ou null}&selling\\_mode={numerico}&token{numerico}](http://url_estabelecimento.com.br/consumption?code={numerico ou null}&selling_mode={numerico}&token{numerico})

Parâmetro	Descrição	Tipo
consumption	Tipo do serviço	String
code	Código de identificação interna da automação (mesa, sala, andar, quarto) No cso de FastFood utilizar NULL	Numérico
selling_mode	Código de identificação interna da autoamação para o modo de venda	String
token	Token validador dos valores citados acima gerados pela automação	String

[Obs. Quando fast food enviar o code como null.](#)

## Retorno

```
"dados":{
  "code":"2/3",
  "open_data":"17/06/2014 17:35:18",
  "itens":[{
    "send":true,
    "serviceValue":0.8,
    "cancelBeforeProduction":false,
    "cancelAfterProduction":false,
    "code":"22",
    "name":"SALADA SUMMER",
    "value":8,
    "amount":1,
    "unit":"UN"
  },
  ],
  "subtotal2Itens":12.6,
  "subtotal2Discount":12.6,
  "subtotal2Servico":12.6,
  "ticketID":"{91EE5EE4-5EF6-E311-8AD9-8851FB3CA0E5}"
}
```

# PEDIDO

Adicionar um item a comanda

POST

[http://url\\_estabelecimento.com.br/pedido/](http://url_estabelecimento.com.br/pedido/)

Parâmetros	Descrição	Tipo
user	Identificação do cliente	String
code	Número de identificação interna da automação Ex: Mesa 3	Numérico
selling_mode	1 = Balcão 2 = Entrega 3 = Mesa 4 = Ficha	Numérico
token	Token gerado pela autoomação - código validador das informações recebidas	String

Exemplo:

```
{
  "user": "Julio"
  "code": "415",
  "itens": [
    {
      "idItem": "11",
      "preco": "5.000000",
      "tipo": "normal",
      "name": "CHEESEBURGUER",
      "quantidade": "1",
      "codigo": "11"
    }
  ],
  "selling_mode": "4"
  "token": "3mlwiu023"
}
```

## Retorno

```
"Response":{  
  Response : True/false  
  Message: {String}  
}
```

## Consulta do saldo devedor – fechamento da conta

Obter valores da ser pago.

GET

[http://url\\_estabelecimento.com.br/cashier?code={2}&selling\\_mode={3}&token{4}](http://url_estabelecimento.com.br/cashier?code={2}&selling_mode={3}&token{4})

Parâmetros	Descrição	Tipo
code	Número de identificação interna da automação Ex: Mesa 3	Numérico
selling_mode	1 = Balcão 2 = Entrega 3 = Mesa 4 = Ficha	Numérico
token	Token gerado pela autoamação - código validador das informações recebidas	String

## Retorno

```
"data":{  
  "consumption":0,  
  "discount":0,  
  "entrance":0,  
  "service":1.26,  
  "subtotal":12.6,  
  "total":13.86  
}
```



# Pagamento

Efetuar pagamento

POST

[http://url\\_estabelecimento.com.br/payment](http://url_estabelecimento.com.br/payment)

Parâmetros	Descrição	Tipo
code	Número de identificação interna da automação Ex: Mesa 3	Numérico
selling_mode	1 = Balcão 2 = Entrega 3 = Mesa 4 = Ficha	Numérico
Way	Código do tipo de pagamento + Valor	Array

Exemplo:

```
{
  "way": [
    {
      "codigo": 8,
      "valor": 30.00
    },
    {
      "codigo": 1,
      "valor": 10.00
    }
  ],
  "selling_mode": "4",
  "code": "415",
  "token": "3mlwui023",
  "consumidor": {
    "cpf_cnpj": "31432040871",
    "nome": "Raphael Rosa",
    "endereco": "End Teste"
  }
}
```

## Retorno

```
"Response":{  
  Response : True/false  
  Message: {String}  
}
```

## Confirmação de pagamento

Verifica se o pagamento foi efetuado pelo Colibri ou Mob2all.

GET

[http://url\\_estabelecimento.com.br/cango?client={1}&sale\\_id={2}](http://url_estabelecimento.com.br/cango?client={1}&sale_id={2})

Parâmetros	Descrição	Tipo
client	ID do Estabelecimento comercial Ex:CNPJ	String
sale_id	Identificação única obtida no serviço de consumo Ex: {9B11CD53-83FC-E311-8AD9-8851FB3CA0E5}	String

## Retorno

```
"data":false/true
```

# Socket

## Acesso ao MENU

Obter o catálogo de produtos por categoria do estabelecimento.

Método Socket

```
{\"funcao\": \"menu\", \"params\": {\"menu\": 1, \"type\": 0, \"}}
```

Parâmetro	Descrição
menu	Tipo do serviço
type	Tipo de menu

## Retorno

Parâmetro	Descrição	Tipo
nameCat	Nome da categoria	String
idCat	identificação da categoria	INT
url_image	Url para exibição de imagens	String
itens	lista de itens	Array
name	nome do item	String
idItem	identificação do item	INT
cost	custo do item	Float
amount	quantidade	Float
order	ordem de views	INT
obs	observações	Array

```
"data": [  
  {  
    "nameCat": "SUCOS",  
    "idCat": 6,  
    "url_image": "https://mob.2all.com.br/imagem....",  
    "itens": [  
      {
```

```

        "name": "Iogurte",
        "idItem": 75,
        "url_image": "https://mob.2all.com.br/imagem...",
        "cost": 0.000000,
        "amount": 1.3,
        "order": 1,
    },
    {
        "name": "GUARANA",
        "idItem": 3,
        "url_image": "https://mob.2all.com.br/imagem...",
        "cost": 2.000000,
        "amount": 0,
        "order": 2,
        "obs": [
            "Com Gelo"
        ]
    }
}
}]

```

## Abertura de Comanda e fast food e consulta

### Pedidos

Obter itens adquiridos.

Método Socket

```
{\ "funcao\": \ "consumption\", \ "params\": {\ "code\": 1, \ "selling_mode\": ABC, \", \ "token\": ABC, \}}
```

Parâmetro	Descrição	Tipo
consumption	Tipo do serviço	String
code	Código de identificação interna da automação (mesa, sala, andar, quarto) No cso de FastFood utilizar NULL	Numérico
selling_mode	Código de identificação interna da autoamação para o modo de venda	String
token	Token validador dos valores citados acima gerados pela automação	String

[Obs. Quando fast food enviar o code como null.](#)

## Retorno

```
"dados":{
  "code":"2/3",
  "open_data":"17/06/2014 17:35:18",
  "itens":[{
    "send":true,
    "serviceValue":0.8,
    "cancelBeforeProduction":false,
    "cancelAfterProduction":false,
    "code":"22",
    "name":"SALADA SUMMER",
    "value":8,
    "amount":1,
    "unit":"UN"
  },
  ],
  "subtotal2Itens":12.6,
  "subtotal2Discount":12.6,
  "subtotal2Servico":12.6,
  "ticketID":"{91EE5EE4-5EF6-E311-8AD9-8851FB3CA0E5}"
}
```

## PEDIDO

Adicionar um item a comanda

Parâmetros	Descrição	Tipo
user	Identificação do cliente	String
code	Número de identificação interna da automação Ex: Mesa 3	Numérico
selling_mode	1 = Balcão 2 = Entrega 3 = Mesa 4 = Ficha	Numérico
token	Token gerado pela autoamação - código validador das informações recebidas	String

MÉTODO: Socket

```
{ "funcao": "order", "params": { "user": "julio", "code": 415, "itens": [{"idItem": "11", "preco": "5.000000", "tipo": "normal", "name": "CHEESE BURGUER", "quantidade": "1", "codigo": "11"}] "selling_mode": "4", "token": "3mlwiu023" }
```

## Retorno

“Response”: {Response: True/false, Message: {String}}

## Consulta do saldo devedor – fechamento da conta

Obter valores da ser pago.

MÉTODO: Socket

```
{ "funcao": "cashier", "params": { "code": 1, "selling_mode": 3, "token": "ABC" }
```

Parâmetros	Descrição	Tipo
code	Número de identificação interna da automação Ex: Mesa 3	Numérico
selling_mode	1 = Balcão 2 = Entrega 3 = Mesa 4 = Ficha	Numérico
token	Token gerado pela autoamação - código validador das informações recebidas	String

## Retorno

```
"data":{
  "consumption":0,
  "discount":0,
  "entrance":0,
  "service":1.26,
  "subtotal":12.6,
  "total":13.86
}
```

## Pagamento

Efetuar pagamento

MÉTODO: Socket

```
{\"funcao\": \"payment\", \"params\": {\"way\": [{\"codigo\": 8, \"valor\": 30.00},
{\"codigo\": 1, \"valor\": 10.00},
{\"selling_mode\": \"4\", \"code\": \"415\", \"token\": \"3mlwiu023\", \"consumidor\":
{\"cpf_cnpj\": \"31432040871\", \"nome\": \"Raphael Rosa\", \"endereco\": \"End Teste\"}}}]}
```

Parâmetros	Descrição	Tipo
code	Número de identificação interna da automação Ex: Mesa 3	Numérico
selling_mode	1 = Balcão 2 = Entrega 3 = Mesa 4 = Ficha	Numérico
Way	Código do tipo de pagamento + Valor	Array

## Retorno

```
"Response": {Response: True/false, Message: {String}}
```

## Confirmação de pagamento

Verifica se o pagamento foi efetuado pelo Colibri ou Mob2all.

MÉTODO: Socket

```
{\"funcao\": \"cango\", \"params\": {\"code\": 1, \"sale_id\": 3, \"}}
```

Parâmetros	Descrição	Tipo
client	ID do Estabelecimento comercial Ex: CNPJ	String
sale_id	Identificação única obtida no serviço de consumo Ex: {9B11CD53-83FC-E311-8AD9-8851FB3CA0E5}	String

## Retorno

“data”: false/true