



Embarcadero Conference

Integrando sua aplicação Delphi/C++ com API do IFOOD

Kleberson Toro

Embarcadero

Conference



O QUE É API

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web.

A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos".



CONCEITO DE API

O conceito de API existe desde os primórdios da computação. Mas nesta palestra iremos falar apenas das web APIs com foco específico na API do “IFood Kitchen”, que é a forma como chamamos os recursos disponibilizados por aplicações web através do protocolo HTTP .

É a troca de mensagens entre aplicações, em sua maioria, estruturadas em formatos XML ou JSON no padrão RESTful.



API RESTful

Oficialmente o termo API RESTful apareceu pela primeira vez na tese de doutorado de Roy Fieldings chamada de “Architectural Styles and the Design of Network-based Software Architectures”, publicada em 2000 pela Universidade da Califórnia. Ali foram descritos os primeiros conceitos sobre API RESTful e muitos são utilizados até hoje.

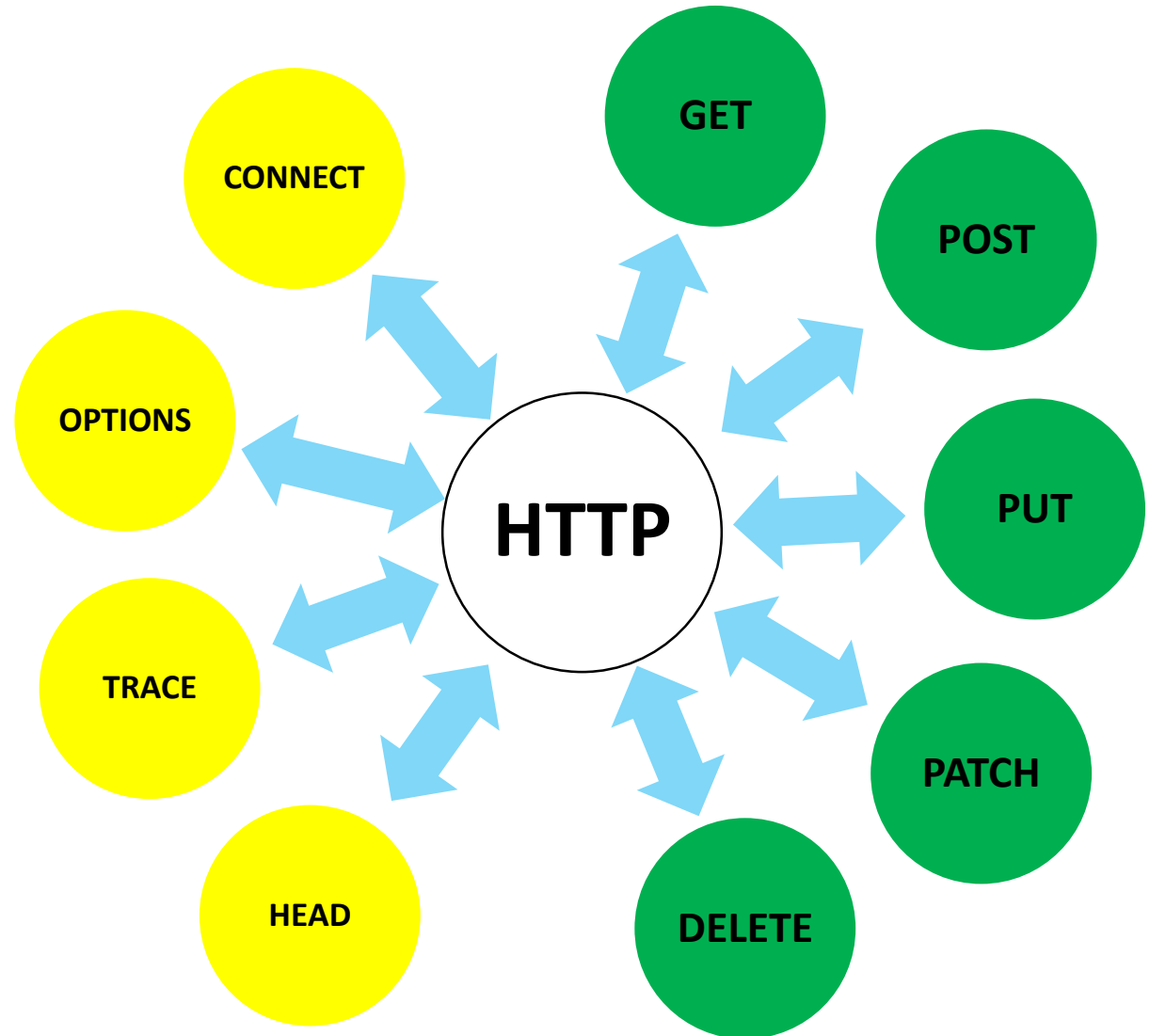
O padrão RESTful conceitua-se por métodos de requisições HTTP.

MÉTODOS DE REQUISIÇÃO HTTP

- GET
- POST
- PUT
- PATCH
- DELETE

Outros métodos, não comumente utilizados:

- CONNECT
- OPTIONS
- TRACE
- HEAD



API DO IFOOD

- Para o uso da API do IFOOD deve-se efetuar um cadastro no portal do “IFood Kitchen” no seguinte link:

<https://developer.ifood.com.br/page/cadastro>

O modelo de autorização da API do IFOOD é o “OAuth2”

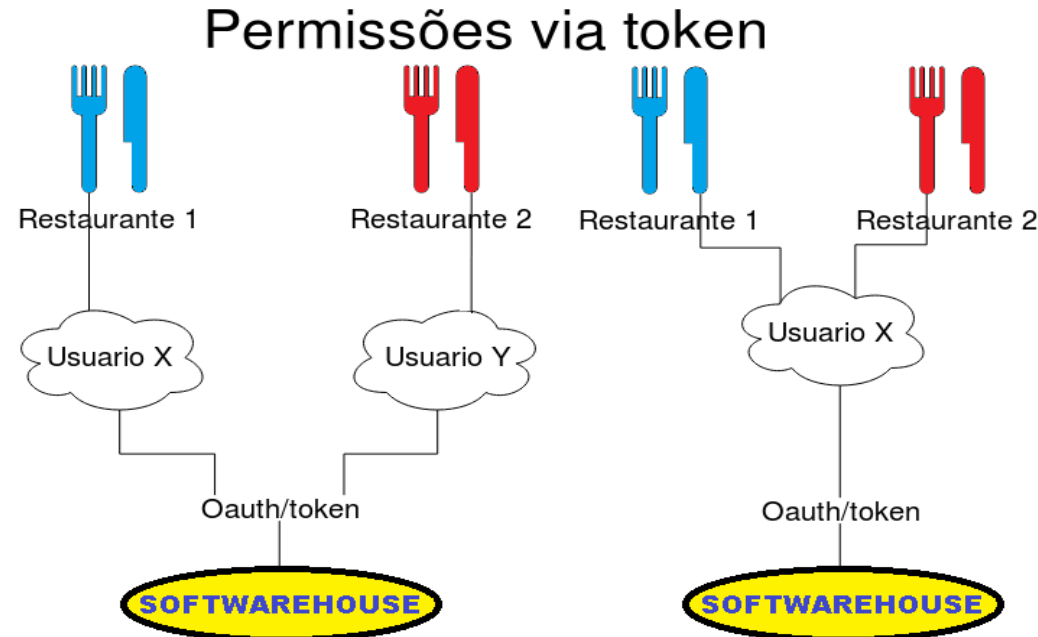
“**client_id**” contem sua identificação para o iFood

“**client_secret**” conterà sua senha

“**grant_type**” é parâmetro obrigatório sempre com a palavra ‘**password**’

“**username**” contem o usuário lojista

“**password**” conterà a senha do usuário lojista



MÉTODOS DA API DO IFOOD

◇ **POST** -/oauth/token

◇ **GET** -/events:polling

POST -/events/acknowledgment

GET -/orders/{reference}

POST -/orders/{reference}/statuses/integration

POST -/orders/{reference}/statuses/confirmation

POST -/orders/{reference}/statuses/dispatch

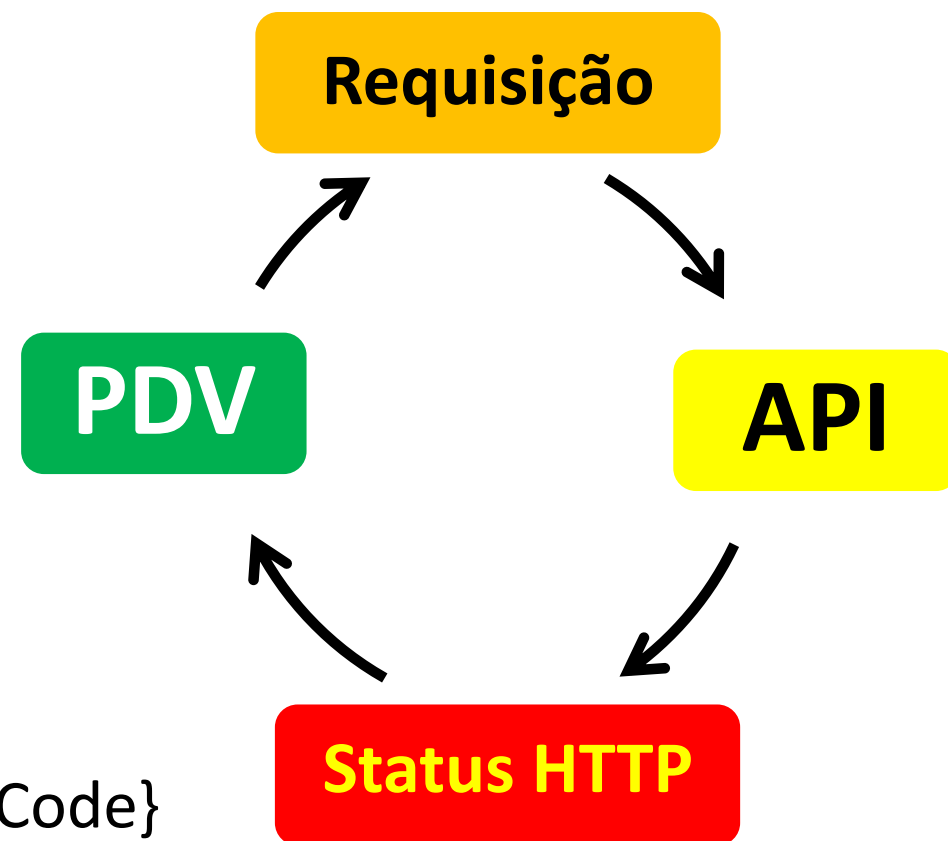
POST -/orders/{reference}/statuses/delivery

POST -/orders/{reference}/statuses/rejection

PATCH -/skus/{externalCode}/prices

PUT -/merchants/{id}/statuses

PATCH -/merchants/{merchantId}/skus/{externalCode}



API do IFOOD – Obtendo o TOKEN

```
1 HTTPClient := TidHTTP.Create;  
2 HTTPClient.Request.Accept := 'application/json';  
3  
4 lParam := TStringList.Create;  
5 lParam.Add('client_id='+edt_client_id.Text); //e-POS id  
6 lParam.Add('client_secret='+edt_client_secret.Text); // e-POS password  
7 lParam.Add('grant_type=password'); //aqui fica escrito 'password' fixo  
8 lParam.Add('username='+edt_username.Text); //merchant's username  
9 lParam.Add('password='+edt_password.Text); //merchant's password  
10  
11 Result := HTTPClient.Post('https://pos-api.ifood.com.br/oauth/token', lParam);
```

✓ Retorno da url: <https://pos-api.ifood.com.br/oauth/token>

```
1 {  
2     "access_token": "eyJ0eXAiOi8qiasZlHW297CbDmE5UvTrJwwF0OjndRfJ2GI",  
3     "token_type": "bearer",  
4     "expires_in": 2539,  
5     "scope": "trust write read"  
6 }
```

API do IFOOD – Método Polling usando TOKEN

```
70 //JsonStreamRetorno: TStringStream
71 JsonStreamRetorno := TStringStream.Create('', TEncoding.UTF8);
72
73 HTTPClient := TidHTTP.Create;
74
75 HTTPClient.Request.Clear;
76 HTTPClient.Request.ContentType := 'application/json';
77 HTTPClient.Request.Charset := 'UTF-8';
78 HTTPClient.Request.BasicAuthentication := false;
79 HTTPClient.Request.CustomHeaders.Clear;
80 HTTPClient.Request.CustomHeaders.AddValue('cache-control', 'no-cache');
81 HTTPClient.Request.CustomHeaders.AddValue('Content-Type', 'application/json');
82 HTTPClient.Request.CustomHeaders.AddValue('Accept', 'application/json');
83 HTTPClient.Request.CustomHeaders.AddValue('Authorization', 'bearer '+aToken); //aqui entra a autorização
84 HTTPClient.Request.CustomHeaders.AddValue('cache-control', 'no-cache');
85 HTTPClient.Request.Accept := 'application/json';
86
87 HTTPClient.get('https://pos-api.ifood.com.br/v1.0/events%3Apolling', JsonStreamRetorno);
88
89 Result := JsonStreamRetorno.DataString; ←
```

API do IFOOD – Retorno do Método Polling

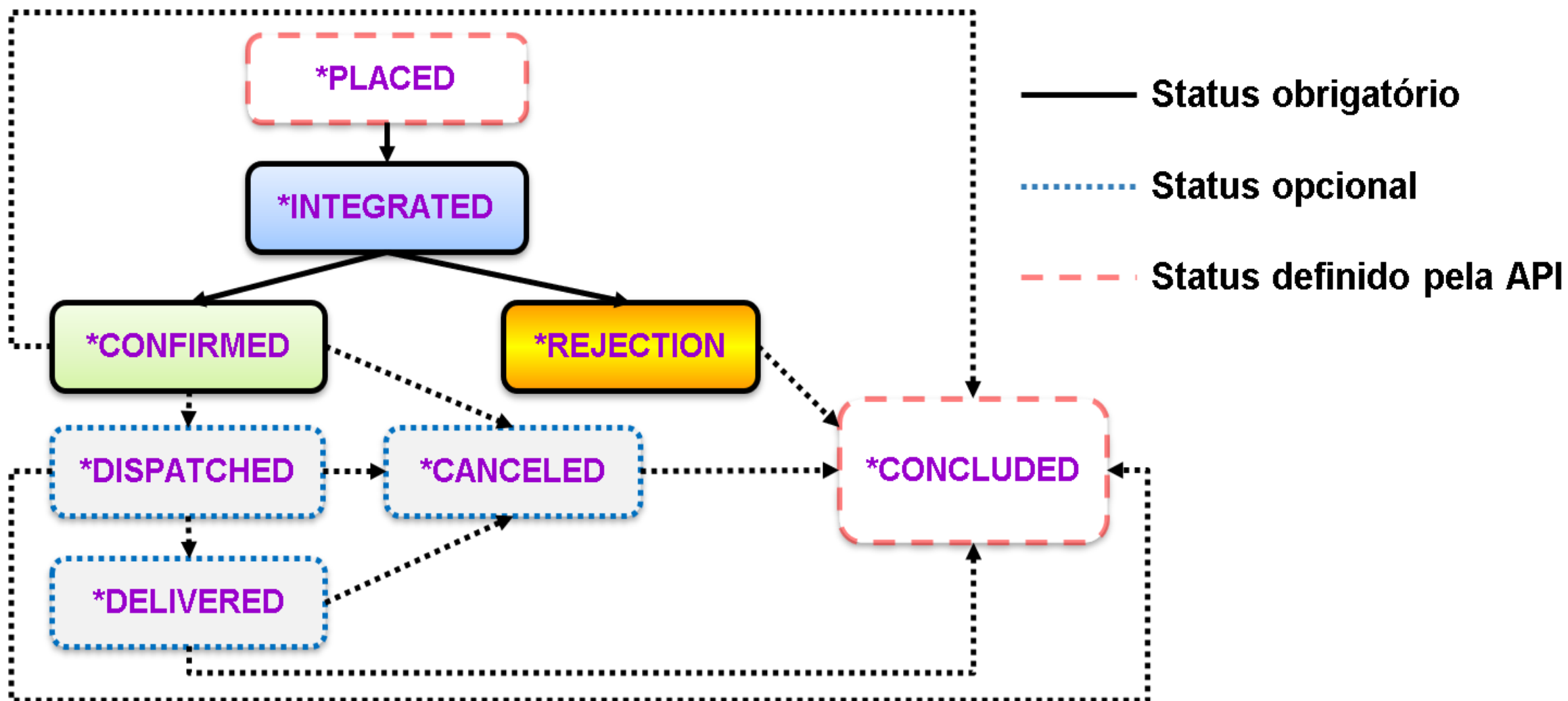
```
1  [
2    {
3      "id": "27734c7a-4789-469a-8a5d-10e98b5237fb",
4      "code": "PLACED",
5      "correlationId": "8101735437170017",
6      "createdAt": "2018-10-21T07:59:17.741Z"
7    },
8    {
9      "id": "27734c7a-4789-469a-8a5d-10e98b5237fb",
10     "code": "PLACED",
11     "correlationId": "9101735417170082",
12     "createdAt": "2018-10-21T08:01:57.741Z"
13   }
14 ]
```

API do IFOOD – Tratando o Retorno do Polling

```
170 //vObj_Array: TJSONArray
171 vObj_Array := TJSONObject.ParseJSONValue(Result) as TJSONArray;
172
173 //vJdsAdapter: TCustomJSONDataSetAdapter
174 vJdsAdapter.Dataset := vFDMemTable;
175 vJdsAdapter.UpdateDataSet( vObj_Array );
176
177 Mem1.Lines.Add('---[ Polling ]---');
178 vFDMemTable.First;
179 while not vFDMemTable.Eof do
180 begin
181     Mem1.Lines.Add('          id: '+vFDMemTable.FieldName('id').AsString);
182     Mem1.Lines.Add('          code: '+vFDMemTable.FieldName('code').AsString);
183     Mem1.Lines.Add('correlationId: '+vFDMemTable.FieldName('correlationId').AsString);
184     Mem1.Lines.Add('      createdAt: '+vFDMemTable.FieldName('createdAt').AsString);
185     Mem1.Lines.Add('---');
186
187     Tb_POLLING.Filtered := False;
188     Tb_POLLING.Filter := 'id = '+QuotedStr(vFDMemTable.FieldName('id').AsString);
189     Tb_POLLING.Filtered := True;
190     if Tb_POLLING.IsEmpty then
191     begin
192         Tb_POLLING.Append;
193         Tb_POLLING.FieldName('id').AsString := vFDMemTable.FieldName('id').AsString;
194         Tb_POLLING.FieldName('code').AsString := vFDMemTable.FieldName('code').AsString;
195         Tb_POLLING.FieldName('correlationId').AsString := vFDMemTable.FieldName('correlationId').AsString;
196         Tb_POLLING.FieldName('createdAt').AsString := vFDMemTable.FieldName('createdAt').AsString;
197         Tb_POLLING.Post;
198     end;
199     vFDMemTable.Next;
200 end;
```

Fluxograma dos Eventos da API do IFOOD

◇ https://pos-api.ifood.com.br/v1.0/orders/{reference}/statuses/*EVENTO





Status da API do IFOOD

- **PLACED** - Indica um pedido foi colocado no sistema.
- **INTEGRATED** - Indica um pedido que foi recebido pelo e-PDV.
- **CONFIRMED** - Indica um pedido confirmado.
- **CANCELLED** - Indica um pedido que foi cancelado.
- **DISPATCHED** - Indica um pedido que foi despachado ao cliente.
- **DELIVERED** - Indica um pedido que foi entregue.
- **CONCLUDED** - Indica um pedido que foi concluído.

OBRIGADO

 kvtoro@gmail.com

 kleberson.toro

  @kvtoro

Pré-Lançamento do curso
Integração com IFOOD com
50% off

<http://bit.ly/integracaoifood>

Embarcadero

Conference



Embarcadero Conference