

Documentação API V2.3

Integracommerce

ÍNDICE

1. INTRODUÇÃO

1.1 API DE HOMOLOGAÇÃO

1.1.1 *Exemplo de chamada via Postman*

1.1.2 *Erro 429*

2. REQUISIÇÕES GET, PUT, POST PARA CADA ENTIDADE INTEGRA

2.1 ATTRIBUTE

2.2 CATEGORY

2.3 MARKETPLACE

2.4 MARKETPLACESTRUCTURE

2.5 PRODUCT

2.6 SKU

2.7 PRICE

2.8 STOCK

2.9 PROMOTION

2.10 ORDER

3. SISTEMA DE FILAS APIINTEGRA

3.1 FILA DE PEDIDOS(ORDERQUEUE)

4. PASSOS PARA HOMOLOGAÇÃO DA API

5. PERGUNTAS FREQUENTES

1. INTRODUÇÃO

Lojas de todos os tamanhos expõem seus produtos em grandes e-commerces, que cobram uma comissão sobre as vendas. É interessante porque todos saem ganhando. O lojista, que alcança milhões de visualizações em seus produtos, podendo aumentar muito suas vendas. Os grandes players, que ampliam e diversificam seu portfólio. E o comprador, que tem mais variedade no site de uma marca que já conhece e confia.

O integracommerce é uma ferramenta que integra os principais marketplaces do mercado ao seu erp, minimizando os custos de integração e dando total liberdade e controle ao lojista da forma mais fácil e automática possível.

A API do Integra Commerce foi criada para automatizar e simplificar o fluxo de informações entre parceiros e nossa plataforma. Ela é utilizada para criar todo o cadastro necessário para expor e vender os produtos nos maiores marketplaces do mercado, cadastrar produtos, gerenciar os estoques e preços e verificar status quando pedidos são feitos e confirmados.

Este documento especifica alguns dos principais requisitos da API Integracommerce. Criado para auxílio aos desenvolvedores, fornecendo as informações necessárias para a execução de seu projeto e implementação, assim como para a realização dos testes e homologação.

As demais seções apresentam a especificação do sistema e estão organizadas como descrito abaixo:

- **Seção 2 – Descrição das chamadas de GET, POST, PUT:** apresenta uma visão geral das chamadas GET, PUT e POST para cada entidade da API.
- **Seção 3 – Modelos de fila da API:** especifica informações sobre as requisições e tratamentos para os modelos de fila da API. Asaber, Order Queue.
- **Seção 4 – Perguntas frequentes:** perguntas que aparecem com frequência ao desenvolver integração via API Integra.

Para mais informações além deste documento acesse:

- Integra API: <https://api.integracommerce.com.br/>
- Integra Swagger Produção: <https://in.integracommerce.com.br/swagger/ui/index>
- Integra Swagger Homologação: <https://api.integracommerce.com.br/swagger/ui/index>

1.1 API em Homologação

Primeiros passos:

- Faça seu cadastro para receber o contato do nosso time com informações para obter acesso ao nosso sandbox e fazer sua integração.
- Leia nossas páginas de padrões de modelagem e navegue no API browser para conhecer as APIs e recursos disponíveis. Depois inicie as integrações com as API's do IntegraCommerce.
- Após a integração inicie o processo de Homologação.
- Nossa API foi construída utilizando REST pelo seu simples entendimento e pode ser adotado em praticamente qualquer cliente ou servidor com suporte a HTTP/HTTPS. Utilizamos o formato JSON para transferência de dados, pois causa uma carga menor na rede.

Para se conectar à API Integracommerce é necessário possuir uma loja para teste em homologação e ainda um usuário API. Pelo qual passarão as requisições.

Utilize o usuário e senha da api para fazer as requisições.

1.1.1 Exemplo de chamada via Postman

Para chamadas via Postman, utilize a Authorization como Basic Auth e informe o usuário e senha da API fornecidos pelo Integra. Conforme figura abaixo.

POST Category

POST Params

Authorization • Headers (2) Body • Pre-request Script Tests Code

Type

Username The authorization header will be generated and added as a custom header

Password Save helper data to request

Show Password

Exemplo Post abaixo.

POST Category

POST Params

Authorization • Headers (2) Body • Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary

```

1 [
2 {
3   "Id": "string",
4   "Name": "string",
5   "ParentId": "string"
6 }
7 ]

```

1.1.2 Erro 429

A API Integracommerce utiliza o erro 429 com a mensagem: Requisição excede a quantidade máxima de chamadas permitidas à API.

Caso receba este erro é necessário tratar o mesmo. De forma que a aplicação deve parar e aguardar por pelo menos 3 segundos até fazer uma nova requisição.

2. REQUISIÇÕES GET, PUT, POST PARA CADA ENTIDADE INTEGRA

Toda a requisição tem que ser originada do parceiro, cada requisição e uma requisição HTTP direcionada para um endpoint.

2.1 Attribute

Atributos são características que acompanham produtos e/ou skus. Exemplo: Cor, tamanho, voltagem, etc.

Na Api Integracommerce é possível recuperar os atributos através do verbo GET.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Attribute>

Em caso de sucesso requisição retornará o(s) resultado(s) com o JSON da seguinte forma:

```
{
  "Page": 0,
  "PerPage": 0,
  "Total": 0,
  "Attributes": [
    {
      "Name": "string"
    }
  ]
}
```

2.2 Category

Categoria é o grupo do qual o produto faz parte.

A categoria é obrigatória para o cadastro de produtos e possui uma estrutura que comporta subcategorias atreladas.

Para cadastrar uma categoria na API Integracommerce utilize o verbo POST.

Chamada POST

URL: <http://api.integracommerce.com.br/api/Category>.

Em casos onde a categoria não será subcategoria, informe o ParentId = "".

No corpo da requisição utilize o JSON abaixo.

```
[
  {
    "Id": "string", // Id da categoria
    "Name": "string", // Nome da categoria
    "ParentId": "string" // Id da categoria Pai
  }
]
```

Obs: Não se esqueça de utilizar os caracteres [] para envolver o corpo do JSON, pois se os mesmos não forem utilizados a API retornará o erro 400 Bad Request com a mensagem 'Valor não pode ser nulo. \n\nNome do parâmetro: source.'

Para cadastrar uma subcategoria na API Integracommerce, utilize o verbo POST.

Chamada POST

URL: <http://api.integracommerce.com.br/api/Category>

No corpo da requisição utilize o JSON abaixo, informando o ParentId igual ao id da categoria que a subcategoria será filha.

```
[
  {
    "Id": "string", // Id da categoria
    "Name": "string", // Nome da categoria
  }
]
```

```
    "ParentId": "string"// Id da categoria pai
  }
]
```

Obs: Não se esqueça de utilizar os caracteres [] para envolver o corpo do JSON, pois se os mesmos não forem utilizados a API retornará o erro 400 Bad Request com a mensagem 'Valor não pode ser nulo. \nNome do parâmetro: source.'

Para recuperar a lista de categorias cadastradas, utilize o verbo GET.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Category>.

Os parâmetros utilizados nesta chamada são:

level: Nível da categoria, sendo o primeiro nível da árvore de categorias o valor0(zero).

page: Página atual.

perPage: Quantidade de resultados por página.

Em caso de sucesso, esta chamada o JSON retornado é:

```
{
  "Page": 0, // Página atual
  "PerPage": 0, // Itens por página
  "Total": 0, // Total de itens
  "Categories": [
    {
      "Id": "string", //Id da categoria
      "Name": "string", // Nome da categoria
      "ParentId": "string" // Id da categoria Pai
    }
  ],
  "CategoriesMarketplace": [
    {
      "MarketPlaceName": "string",// Nome do Marketplace
      "CategoryMarketplaceList": {
        "CategoryId": 0, //Id categoria
        "CategoryName": "string", // Nome categoria
        "SubCategoryId": 0, // Id da subcategoria
        "SubCategoryName": "string", // Nome da subcategoria
        "FamilyId": 0, // Id da familia da categoria
        "FamilyName": "string", // Nome da familia da cateogria
        "SubFamilyId": 0, //Id da sub familia da categoria
        "SubFamilyName": "string" // Nome da sub familia da categoria
      }
    }
  ]
}
```

O Objeto **CategoriesMarketplace** serve para o caso de uma categoria no Integra ter um link com uma categoria de algum Marketplace específico. Caso não utilize, este objeto virá em branco nesta chamada.

Para recuperar uma categoria específica use o verbo GET.

Chamada GET.

URL: <http://api.integracommerce.com.br/api/Category/api/Category/{Id}>

O parâmetro utilizado nesta requisição é o Id da categoria buscada.

Em caso de sucesso, esta chamada retorna o seguinte JSON

```
{
  "Id": "string", // Id da categoria
  "Name": "string", // Nome da categoria
  "ParentId": "string" // Id categoria pai
}
```

2.3 Marketplace

Este método é utilizado para recuperar todos os *Marketplaces* que integram com o Integracommerce.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Marketplace>

Em caso de sucesso, o JSON de resposta será:

```
{
  "Marketplaces": [
    {
      "Name": "string" // Nome do Marketplace
    }
  ]
}
```

2.4 MarketplaceStructure

Este método é utilizado para recuperar todas as categorias dos *Marketplaces* que estão no Integracommerce.

Chamada GET

URL: <http://api.integracommerce.com.br/api/MarketplaceStructure>

Utilizando como parâmetros:

marketplaceName: Nome do marketplace

parentId: Id da categoria pai. O primeiro nível da árvore de categorias é 0(zero).

page: Página atual da requisição.

perPage: Número de itens por página.

Em caso de sucesso, a resposta será:

```
{
  "Page": 0, // Página
  "PerPage": 0, // Quantidade por página
}
```

```

"Total": 0, // Total de categorias encontradas
"MarketplaceStructures": [
  {
    "Id": "string", // Id da categoria
    "Name": "string", // Nome da categoria
    "ParentId": "string" // Id da categoria pai
  }
]
}

```

2.5 Product

Para a criação de produtos utilize o verbo POST. O objeto referente a atributos não é obrigatório. Apenas declare o objeto passando o mesmo sem seus dados da seguinte forma: Attributes: []

Chamada POST

URL: <http://api.integracommerce.com.br/api/Product>

No corpo da Requisição utilize o JSON:

```

{
  "IdProduct": "string", // Id do produto (Obrigatório)
  "Name": "string", // Nome do produto (Obrigatório)
  "Code": "string", // Código do produto no ERP, pode ser igual ao
  IdProduct (Obrigatório)
  "Brand": "string", // Marca (Obrigatório)
  "NbmOrigin": "string", // NBM de origem, 0 para nacional, 1 para
  importado. (Obrigatório)
  "NbmNumber": "string", // NBM é o mesmo que o NCM do produto
  (Opcional)
  "WarrantyTime": "string", // Tempo de garantia em meses (Obrigatório)
  "Active": true, // Status do produto (Obrigatório)
  "Categories": [ // (Obrigatório) Quantidade máxima de registros: 4.
  Caso ultrapasse o limite, serão considerados apenas os 4 primeiros
  registros.
    {
      "Id": "string", // Id da categoria
      "Name": "string", // Nome da categoria
      "ParentId": "string" // Id da categoria pai
    }
  ],
  "Attributes": [ // (Obrigatório)
    {
      "Name": "string", // Nome do atributo
      "Value": "string" // Valor do atributo
    }
  ]
}

```

Chamada GET

URL: <http://api.integracommerce.com.br/api/Product/api/Product>

Utilizando como parâmetros:

page: Página atual da requisição.

perPage: Número de itens por página.

Em caso de sucesso, a resposta será:

```
{
  "Page": 0, // Página atual
  "PerPage": 0, // Número de registros por página
  "Total": 0, //Total de produtos
  "Products": [
    {
      "IdProduct": "string", // Id do produto
      "Name": "string", // Nome do produto
      "Code": "string", //Código do produto no ERP
      "Brand": "string", // Marca
      "NbmOrigin": "string", // NBM de origem, 0 para nacional, 1 para
      importado.
      "NbmNumber": "string", //NBM é o mesmo que o NCM do produto
      "WarrantyTime": "string", // Tempo de garantia em meses
      "Active": true, // Status do produto
      "Categories": [
        {
          "Id": "string", // Id da categoria
          "Name": "string", // Nome da categoria
          "ParentId": "string" // Id da categoria pai
        }
      ],
      "Attributes": [
        {
          "Name": "string", // Nome do atributo
          "Value": "string" // Valor do atributo
        }
      ]
    }
  ]
}
```

Para pesquisar por um Produto específico utilize o verbo GET.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Product/{Id}>

Utilizando como parâmetro o Id do produto buscado.

Em caso de sucesso o seguinte JSON será retornado:

```
{
```

```

    "IdProduct": "string", // Id do produto
    "Name": "string", // Nome do produto
    "Code": "string", //Código do produto no ERP
    "Brand": "string", // Marca
    "NbmOrigin": "string", // NBM de origem, 0 para nacional, 1 para
importado.
    "NbmNumber": "string", //NBM é o mesmo que o NCM do produto
    "WarrantyTime": "string", // Tempo de garantia em meses
    "Active": true, // Status do produto
    "Categories": [
      {
        "Id": "string", // Id da categoria
        "Name": "string", // Nome da categoria
        "ParentId": "string" // Id da categoria pai
      }
    ],
    "Attributes": [
      {
        "Name": "string", // Nome do atributo
        "Value": "string" // Valor do atributo
      }
    ]
  }
}

```

Para atualização de um produto utilize o verbo PUT. Em casos de atualização do campo Active para *true* ou para *false*, os Skus relacionados ao produto não terão alteração em seus status.

Chamada PUT

URL: <http://api.integracommerce.com.br/api/Product>

Como corpo da requisição utilize o JSON:

```

{
  "IdProduct": "string", // Id do produto
  "Name": "string", // Nome do produto
  "Code": "string", //Código do produto no ERP
  "Brand": "string", // Marca
  "NbmOrigin": "string", // NBM de origem, 0 para nacional, 1 para
importado.
  "NbmNumber": "string", //NBM é o mesmo que o NCM do produto
  "WarrantyTime": "string", // Tempo de garantia em meses
  "Active": true, // Status do produto
  "Categories": [ //Quantidade máxima de registros: 4. Caso ultrapasse
o limite, serão considerados apenas os 4 primeiros registros.
    {
      "Id": "string", // Id da categoria
      "Name": "string", // Nome da categoria
      "ParentId": "string" // Id da categoria pai
    }
  ],
}

```

```

    "Attributes": [
      {
        "Name": "string", // Nome do atributo
        "Value": "string" // Valor do atributo
      }
    ]
  }
}

```

2.6 Sku

Para a criação de skus utilize o verbo POST.

Chamada POST

URL: <http://api.integracommerce.com.br/api/Sku>

O objeto referente a atributos não é obrigatório. Apenas declare o objeto passando o mesmo sem seus dados da seguinte forma: Attributes: []

As urls das imagens devem ser informadas utilizando o protocolo http. Não deve ser utilizado para estas o protocolo https.

Caso não seja definida uma imagem principal, a primeira imagem será considerada como principal. A url também não deve conter espaços.

O corpo do JSON da requisição será:

```

{
  "IdSku": "string", // Id do Sku (Obrigatório)
  "IdSkuErp": "string", // Id do Sku no ERP (Obrigatório)
  "IdProduct": "string", // Id do produto (Obrigatório)
  "Name": "string", // Nome do Sku (Obrigatório)
  "Description": "string", // Descrição do Sku (Obrigatório)
  "Height": 0, // Dimensão de altura em metros (Obrigatório)
  "Width": 0, // Dimensão de largura em metros (Obrigatório)
  "Length": 0, // Dimensão de comprimento em metros (Obrigatório)
  "Weight": 0, // Peso em Kg (Obrigatório)
  "CodeEan": "string", // EAN (Obrigatório)
  "CodeNcm": "string", // NCM (Opcional)
  "CodeIsbn": "string", // ISBN em caso do produto ser livro. Campo deve
  possuir 10 ou 13 caracteres. (Opcional)
  "CodeNbm": "string", // NBM do Sku (Opcional)
  "Variation": "string", // Variação (Opcional)
  "Status": true, // Status do Sku (Obrigatório)
  "Price": { (Obrigatório)
    "ListPrice": 0, // Preço de
    "SalePrice": 0 // Preço por
  },
  "StockQuantity": 0, // Quantidade em estoque (Obrigatório)
  "MainImageUrl": "string", //Adiciona a imagem como principal(Opcional)
  "UrlImages": [ // Urls das imagens do Sku (Opcional). Usar http e não
  https
    "string"
  ],
}

```

```
"Attributes": [ // Atributos do Sku (Opcional)
  {
    "Name": "string", // Nome do atributo
    "Value": "string" // Valor do Atributo
  }
]
```

Os campos opcionais, não devem ser omitidos, apenas passados em branco("") caso não queira preenchê-lo.

Para recuperar todos skus cadastrados utilize o verbo GET.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Sku>

Utilizando como parâmetros:

page: Página atual da requisição.

perPage: Número de itens por página.

O JSON de resposta em caso de sucesso será:

```
{
  "Page": 0, // Página atual
  "PerPage": 0, // Itens por página
  "Total": 0, // Total de itens
  {
    "Skus": [
      {
        "IdSku": "string", // Id do Sku
        "IdSkuErp": "string", // Id do Sku no ERP
        "IdProduct": "string", // Id do produto
        "Name": "string", // Nome do Sku
        "Description": "string", // Descrição do Sku
        "Height": 0, // Dimensão de altura em metros
        "Width": 0, // Dimensão de largura em metros
        "Length": 0, // Dimensão de comprimento em metros
        "Weight": 0, // Peso em Kg
        "CodeEan": "string", // EAN
        "CodeNcm": "string", // NCM
        "CodeIsbn": "string", // ISBN em caso do produto ser livro
        "CodeNbm": "string", // NBM do Sku
        "Variation": "string", // Variação
        "Status": true, // Status do Sku
        "Price": {
          "ListPrice": 0, // Preço de
          "SalePrice": 0 // Preço por
        },
        "StockQuantity": 0, // Quantidade em estoque,
        "MainImageUrl": "string", // Url da imagem principal
        "UrlImages": [ // Urls das imagens do Sku
```

```

    "string"
  ],
  "Attributes": [ // Atributos do Sku
    {
      "Name": "string", // Nome do atributo
      "Value": "string" // Valor do Atributo
    }
  ]
}
]
}

```

Para pesquisar por um Sku específico utilize o verbo GET.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Sku/{Id}>

Utilizando como parâmetro o Id do Sku buscado.

Em caso de sucesso o seguinte JSON será retornado:

```

{
  "IdSku": "string", // Id do Sku
  "IdSkuErp": "string", // Id do Sku no ERP
  "IdProduct": "string", // Id do produto
  "Name": "string", // Nome do Sku
  "Description": "string", // Descrição do Sku
  "Height": 0, // Dimensão de altura em metros
  "Width": 0, // Dimensão de largura em metros
  "Length": 0, // Dimensão de comprimento em metros
  "Weight": 0, // Peso em Kg
  "CodeEan": "string", // EAN
  "CodeNcm": "string", // NCM
  "CodeIsbn": "string", // ISBN em caso do produto ser livro
  "CodeNbm": "string", // NBM do Sku
  "Variation": "string", // Variação
  "Status": true, // Status do Sku
  "Price": {
    "ListPrice": 0, // Preço de
    "SalePrice": 0 // Preço por
  },
  "StockQuantity": 0, // Quantidade em estoque
  "MainImageUrl": "string", //Url da imagem principal
  "UrlImages": [ // Urls das imagens do Sku
    "string"
  ],
  "Attributes": [ // Atributos do Sku
    {
      "Name": "string", // Nome do atributo
      "Value": "string" // Valor do Atributo
    }
  ]
}

```

```
    }
  ]
}
```

Para atualização de um Sku utilize o verbo PUT. Para atualização apenas do Status, o JSON inteiro do produto deve ser informado, mesmo que o Sku não possua alterações.

Chamada PUT

URL: <http://api.integracommerce.com.br/api/Sku>

O corpo do JSON da requisição será:

```
{
  "IdSku": "string", // Id do Sku
  "IdSkuErp": "string", // Id do Sku no ERP
  "IdProduct": "string", // Id do produto
  "Name": "string", // Nome do Sku
  "Description": "string", // Descrição do Sku
  "Height": 0, // Dimensão de altura em metros
  "Width": 0, // Dimensão de largura em metros
  "Length": 0, // Dimensão de comprimento em metros
  "Weight": 0, // Peso em Kg
  "CodeEan": "string", // EAN
  "CodeNcm": "string", // NCM
  "CodeIsbn": "string", // ISBN em caso do produto ser livro
  "CodeNbm": "string", // NBM do Sku
  "Variation": "string", // Variação
  "Status": true, // Status do Sku
  "Price": {
    "ListPrice": 0, // Preço de
    "SalePrice": 0 // Preço por
  },
  "StockQuantity": 0, // Quantidade em estoque
  "MainImageUrl": "string", //Url da Imagem principal
  "UrlImages": [ // Urls das imagens do Sku
    "string"
  ],
  "Attributes": [ // Atributos do Sku
    {
      "Name": "string", // Nome do atributo
      "Value": "string" // Valor do Atributo
    }
  ]
}
```

Ao ser atualizado, o Sku será inserido na SkuQueue

2.7 Price

Para atualizar o preço de um Sku utilize o verbo PUT

Chamada PUT

URL: <http://api.integracommerce.com.br/api/Price>

Quantidade de Itens por Request: 100

Utilize como corpo da requisição o seguinte JSON:

```
[
  {
    "IdSku": "string", // Id do sku
    "ListPrice": 0, //preço De:
    "SalePrice": 0 // Preço por
  }
]
```

Obs: Não se esqueça de utilizar os caracteres [], pois se os mesmos não forem utilizados a API retornará o erro 400 Bad Request com a mensagem 'Referência de objeto não definida para uma instância de um objeto.'

2.8 Stock

Para atualizar o estoque de um Sku utilize o verbo PUT.

Chamada PUT

URL: <http://api.integracommerce.com.br/api/Stock>

Quantidade de Itens por Request: 100

Utilize como corpo da requisição o seguinte JSON:

```
[
  {
    "IdSku": "string", // Id do Sku
    "Quantity": 0 // Quantidade
  }
]
```

Ao ser atualizado, o estoque será inserido na StockQueue.

Obs: Não se esqueça de utilizar os caracteres [], pois se os mesmos não forem utilizados a API retornará o erro 400 Bad Request com a mensagem 'Referência de objeto não definida para uma instância de um objeto.'

2.9 Promotion

Para criação de promoções utilize o verbo POST

Chamada POST

URL: <http://api.integracommerce.com.br/api/Promotion>

Utilize como corpo da requisição o seguinte JSON:

```
{
  "IdSku": "string", // Id do Sku
```

```
"ListPrice": 0, // Preço de
"SalePrice": 0, // Preço por
"StartDate": "string", //Data de início da promoção
"EndDate": "string" //Data de término da promoção
}
```

2.10 Order

Para recuperar todos pedidos cadastrados utilize o verbo GET

Chamada GET

URL: <http://api.integracommerce.com.br/api/Order>

Utilizando como parâmetros:

page: Página atual da requisição.

perPage: Número de itens por página.

Status: Status do pedido

O JSON de resposta em caso de sucesso será:

```
{
  "Page": 0,
  "PerPage": 0,
  "Total": 0,
  "Orders": [
    {
      "OrderStatus": "string",
      "IdOrder": "string",
      "EstimatedDeliveryDate": "2021-01-28T21:41:10.172Z",
      "TotalAmount": "string",
      "TotalFreight": "string",
      "TotalDiscount": "string",
      "ApprovedDate": "2021-01-28T21:41:10.172Z",
      "ReceiverName": "string",
      "CustomerMail": "string",
      "TotalTax": "string",
      "IdOrderMarketplace": "string",
      "InsertedDate": "2021-01-28T21:41:10.172Z",
      "PurchasedDate": "2021-01-28T21:41:10.172Z",
      "UpdatedDate": "2021-01-28T21:41:10.172Z",
      "MarketplaceName": "string",
      "StoreName": "string",
      "UpdatedMarketplaceStatus": true,
      "InsertedErp": true,
      "CustomerPfCpf": "string",
      "CustomerPfName": "string",
      "CustomerPjCnpj": "string",
      "CustomerPjCorporatename": "string",
      "DeliveryAddressStreet": "string",
      "DeliveryAddressAdditionalInfo": "string",
      "DeliveryAddressZipcode": "string",
      "DeliveryAddressNeighborhood": "string",
      "DeliveryAddressCity": "string",
      "DeliveryAddressReference": "string",
    }
  ]
}
```

```

"DeliveryAddressState": "string",
"DeliveryAddressNumber": "string",
"TelephoneMainNumber": "string",
"TelephoneSecondaryNumber": "string",
"TelephoneBusinessNumber": "string",
"CustomerBirthDate": "string",
"CustomerPjIe": "string",
"InvoicedNumber": "string",
"InvoicedLine": 0,
"InvoicedIssueDate": "2021-01-28T21:41:10.172Z",
"InvoicedKey": "string",
"InvoicedDanfeXml": "string",
"ShippedTrackingUrl": "string",
"ShippedTrackingProtocol": "string",
"ShippedEstimatedDelivery": "2021-01-28T21:41:10.172Z",
"ShippedCarrierDate": "2021-01-28T21:41:10.172Z",
"ShippedCarrierName": "string",
"ShipmentExceptionObservation": "string",
"ShipmentExceptionOccurrenceDate": "2021-01-28T21:41:10.172Z",
"DeliveredDate": "2021-01-28T21:41:10.172Z",
"ShippedCodeERP": "string",
"BranchDocument": "string",
"MarketplaceDocument": "string",
"SellerName": "string",
"GatewayPaymentsDocument": "string",
"Products": [
  {
    "IdSku": "string",
    "Quantity": 0,
    "Price": "string",
    "Freight": "string",
    "Discount": "string",
    "IdOrderPackage": 0
  }
],
"Payments": [
  {
    "Name": "string",
    "Installments": 0,
    "Amount": 0
  }
]
}
]
}

```

Considerações sobre o campo “Payments”

O campo Payments pode retornar um ou mais meios de pagamento. O cliente pode pagar o pedido com mais de uma forma de pagamento. Atualmente as formas de pagamentos disponíveis são:

- Boleto Bancário → bank_slip
- MagaluPay → account_balance
- Cartão de Crédito → credit_card
- Cartão de Débito → debit_card
- Pix → pix

Considerações sobre os campos "Discount", "Freight" e "Price"

Se houver duas ou mais quantidades de um mesmo Sku, tanto o "Price" como "Discount" é sobre o valor unitário.

Exemplo:

SKU: Produto A

Qtd: 2

Valor unitário: 100,00

Valor unitário (Price): R\$ 100,00 – R\$ 5,00 = R\$ 95,00

Desconto (Discount): R\$ 5,00

Valor total do pedido = $([2 \times 100] - [2 \times 5]) = 190$

Já o frete(freight) é em relação ao número total de Skus.

Exemplo:

SKU: Produto A

Qtd: 2

Valor unitário (Price): R\$ 100,00 – R\$ 5,00 = R\$ 95,00

Desconto (Discount): R\$ 5,00

Frete (Freight): 15,00

SKU: Produto B

Qtd: 1

Valor unitário (Price): R\$ 50,00 – R\$ 10,00 = R\$ 40,00

Desconto (Discount): R\$ 10,00

Frete (Freight): 5,00

Frete total = 20,00

Valor total do Pedido = $([2 \times 100] - [2 \times 5] + 15,00(\text{frete A}) + [50] - [10] + 5,00(\text{frete B})) = 205 + 45 = 250$

Para pesquisar por um pedido específico utilize o verbo GET.

Chamada GET

URL: <http://api.integracommerce.com.br/api/Order/{Id}>

Utilizando como parâmetro o Id do pedido buscado.

Em caso de sucesso o seguinte JSON será retornado:

```
{
  "OrderStatus": "string",
  "IdOrder": "string",
  "EstimatedDeliveryDate": "2021-01-28T21:41:10.156Z",
  "TotalAmount": "string",
  "TotalFreight": "string",
  "TotalDiscount": "string",
  "ApprovedDate": "2021-01-28T21:41:10.156Z",
  "ReceiverName": "string",
  "CustomerMail": "string",
  "TotalTax": "string",
  "IdOrderMarketplace": "string",
  "InsertedDate": "2021-01-28T21:41:10.156Z",
  "PurchasedDate": "2021-01-28T21:41:10.156Z",
  "UpdatedDate": "2021-01-28T21:41:10.156Z",
  "MarketplaceName": "string",
  "StoreName": "string",
  "UpdatedMarketplaceStatus": true,
  "InsertedErp": true,
  "CustomerPfCpf": "string",
  "CustomerPfName": "string",
  "CustomerPjCnpj": "string",
  "CustomerPjCorporatename": "string",
  "DeliveryAddressStreet": "string",
  "DeliveryAddressAdditionalInfo": "string",
  "DeliveryAddressZipcode": "string",
  "DeliveryAddressNeighborhood": "string",
```

```

"DeliveryAddressCity": "string",
"DeliveryAddressReference": "string",
"DeliveryAddressState": "string",
"DeliveryAddressNumber": "string",
"TelephoneMainNumber": "string",
"TelephoneSecondaryNumber": "string",
"TelephoneBusinessNumber": "string",
"CustomerBirthDate": "string",
"CustomerPjIe": "string",
"InvoicedNumber": "string",
"InvoicedLine": 0,
"InvoicedIssueDate": "2021-01-28T21:41:10.156Z",
"InvoicedKey": "string",
"InvoicedDanfeXml": "string",
"ShippedTrackingUrl": "string",
"ShippedTrackingProtocol": "string",
"ShippedEstimatedDelivery": "2021-01-28T21:41:10.156Z",
"ShippedCarrierDate": "2021-01-28T21:41:10.156Z",
"ShippedCarrierName": "string",
"ShipmentExceptionObservation": "string",
"ShipmentExceptionOccurrenceDate": "2021-01-28T21:41:10.156Z",
"DeliveredDate": "2021-01-28T21:41:10.156Z",
"ShippedCodeERP": "string",
"BranchDocument": "string",
"MarketplaceDocument": "string",
"SellerName": "string",
"GatewayPaymentDocument": "string",
"Products": [
  {
    "IdSku": "string",
    "Quantity": 0,
    "Price": "string",
    "Freight": "string",
    "Discount": "string",
    "IdOrderPackage": 0
  }
],
"Payments": [
  {
    "Name": "string",
    "Installments": 0,
    "Amount": 0
  }
]
}

```

Ao atualizar o status do pedido os seguintes Status devem ser utilizados: INVOICED, SHIPPED, DELIVERED, SHIPMENTEXCEPTION, CANCELED. Cada Status atualizado deve seguir uma ordem crescente, ou seja, se o Status está em Shipped(Despachado) o mesmo não pode retornar para um status anterior, como Invoiced(Nota Fiscal Emitida).

Os pedidos com Status NEW (aguardando pagamento) não precisam ter o status alterado para Processing. O status PROCESSING configura um pedido já aprovado e em processo de separação para envio ao cliente.

Para confirmar o recebimento de um pedido da OrderQueue é necessário passar apenas o Id da fila. Não é preciso informar que o pedido está com Status diferente. Verificar JSON de PUT na OrderQueue.

Quando o pedido é aprovado ou cancelado ele vem para a OrderQueue novamente com Status APPROVED ou CANCELED respectivamente. Caso ele esteja APPROVED, utilize o campo ApprovedDate e alterar seu Status para PROCESSING. E novamente faça um post do Id da fila na Order - Queue confirmando seu consumo.

No status NEW o pedido ainda não foi aprovado. Desta forma não há ainda movimentação em esto - que, o Integra possui um sistema de reserva de estoque. Este pode ser configurado diretamente no painel. A configuração deve conter quantos dias o produto ficará na reserva. Até que um pedido seja realmente pago. Só então o estoque é realmente descontado.

Caso cliente não possa trabalhar dessa forma. É necessário trabalhar apenas com pedidos APPROVED. Deixando os pedidos em Status NEW de lado. E inserir no ERP ou em outros sistemas, apenas quando os pedidos forem Aprovados.

Os status de NEW para APPROVED ou CANCELED são atualizados automaticamente no Integra, conforme o Marketplace informa esses novos status.

Para atualização de um pedido utilize o verbo PUT

Chamada PUT

URL: <http://api.integracommerce.com.br/api/Order>

Como corpo da requisição utilize o JSON, para cada Status possível:

Em caso de sucesso a resposta será 200 OK

JSON INVOICED (Nota fiscal emitida):

```
{
  "IdOrder": "string", // Id do pedido
  "OrderStatus": "INVOICED", // Status do pedido
  "InvoicedNumber": "string", // Número da nota fiscal
  "InvoicedLine": 0, // Serie da nota fiscal
  "InvoicedIssueDate": "2016-06-23T10:59:50.381Z", // Data de atualização
  "InvoicedKey": "string", // Chave da nota fiscal
  "InvoicedDanfeXml": "string" // XML da nota fiscal
}
```

JSON SHIPPED(Pedido despachado):

```
{
  "IdOrder": "string", // Id do pedido
  "OrderStatus": "SHIPPED", // Status do pedido
  "ShippedTrackingUrl": "string", // Url de rastreio
  "ShippedTrackingProtocol": "string", // Código de rastreio do pedido
  "ShippedEstimatedDelivery": "2016-06-23T10:59:50.381Z", // Data estimada
  de entrega
  "ShippedCarrierDate": "2016-06-23T10:59:50.381Z", // Data de despacho do
  pedido
  "ShippedCarrierName": "string" // Nome da transportadora
}
```

JSON DELIVERED(Pedido Entregue):

```
{
  "IdOrder": "string", // Id do pedido
```

```
"OrderStatus": "DELIVERED", // Status do pedido
"DeliveredDate": "2016-06-23T10:59:50.381Z" // Data de entrega
}
```

JSON SHIPMENTEXCEPTION (Exceção de entrega):

```
{
  "IdOrder": "string", // Id do pedido
  "OrderStatus": "SHIPMENTEXCEPTION", // Status do pedido
  "ShipmentExceptionObservation": "string", // Observação
  "ShipmentExceptionOccurrenceDate": "2016-06-23T10:59:50.381Z" // Data da
  ocorrência
}
```

JSON CANCELED(Pedido cancelado):

```
{
  "IdOrder": "string", // Id do pedido
  "OrderStatus": "CANCELED" // Status do pedido
}
```

Ao ser atualizado, o pedido será inserido na OrderQueue.

Para gerar etiquetas de pedidos Magalu Entregas utilizar o verbo POST

Chamada POST

URL: <http://api.integracommerce.com.br/api/Order/ShippingLabels>

Utilize como corpo da requisição o seguinte JSON:

```
{
  "Format": "string", //Formato do(s) arquivos
  "Orders": [
    "string" //Lista de pedidos
  ]
}
```

O JSON de resposta em caso de sucesso será:

```
[
  {
    "Url": "https://...", //url para download das etiquetas
    "ExpiresOn": "2099-12-12T15:06:31.577597+00:00", //Expiração do link
    "Orders": [
      {
        "Order": "string",
        "TrackingCode": "string", //Código de rastreo do pedido
        "TrackingUrl": "http://..." //Url de rastreo
      }
    ]
  }
]
```

3. SISTEMA DE FILAS API INTEGRA

O sistema de filas tem como objetivo facilitar a leitura de inserções e atualizações de itens da API Integra.

Como todo sistema de fila, o primeiro a chegar é o primeiro a sair, então considerando que as filas têm um limite de 100 itens, somente irão aparecer novos itens à medida que os itens da fila são confirmados (consumidos).

Para consumir os itens da fila é necessário enviar os Ids da OrderQueue. Deve-se atentar que não se trata do Id do SKU ou do ORDER. Todas as filas seguem o mesmo padrão de consumo através de uma chamada PUT, onde passa como parâmetro a lista de Ids, como no exemplo abaixo:

Chamada: `PUT https://api-integra.azurewebsites.net/api/OrderQueue`

Body: `[{ "Id": 1228 }]`

O retorno em caso de sucesso é 204 (No content), caso contrário é retornado apenas os itens que não foram possíveis a confirmação, os demais serão confirmados (consumidos).

3.1 Fila de pedidos (OrderQueue)

Nessa fila serão disponibilizados os novos pedidos (NEW) e as alterações de status de pedido que vieram do Marketplace (APPROVED, CANCELED).

Ao fazer o GET dos pedidos com status APPROVED e confirmar o consumo destes pedidos na OrderQueue, é necessário um novo PUT, mas desta vez em Orders atualizando o status do pedido de APPROVED para PROCESSING.

Chamada: `https://api-integra.azurewebsites.net/api/api/OrderQueue`

Exemplo de retorno:

```
{
  "Total": 1,
  "OrderQueues": [
    {
      "Id": 1228,
      "IdOrder": "262203273203",
      "IdOrderMarketplace": "02-622032732",
      "InsertedDate": "01-11-2016",
      "OrderStatus": "NEW"
    }
  ]
}
```

4. PASSOS PARA HOMOLOGAÇÃO DA API

Para que a API esteja homologada corretamente é necessário verificar os seguintes passos abaixo. Em caso de dúvidas entre em contato com a equipe Integracommerce.

- 1.1 Foi criado tratamento correto para o erro 429?
- 1.2 ID de um item simples que tenha sido cadastrado
- 1.3 ID de um item com mais de um sku que tenha sido cadastrado por vocês
- 1.4 ID de um item que tenha seu preço alterado
- 1.5 ID de um item que tenha seu estoque alterado

- 1.6 Atualização de Pedidos
 - 1.6.1 ID de um Pedido atualizado para o status – PROCESSING
 - 1.6.2 ID de um Pedido atualizado para o status – INVOICED
 - 1.6.3 ID de um Pedido atualizado para o status – SHIPPED
 - 1.6.4 ID de um Pedido atualizado para o status – DELIVERED
 - 1.6.5 ID de um Pedido atualizado para o status – SHIPMENT_EXCEPTION
- 1.7 Sabendo-se que alguns marketplaces realizam a validação do Cálculo do dígito verificador EAN 13 no cadastro de produtos, qual a solução encontrada para evitar o envio de EAN inválido?
 - 1.7.1 Em caso de skus sem EAN, é possível enviar sku sem EAN?
- 1.8 As informações sobre dimensão do SKU não são obrigatórias mas se enviadas, não devem ser menor ou igual à zero. Qual tratamento para consistir esta informação antes do envio?
- 1.9 Existe alguma impossibilidade de atualização para algum dos nossos status de pedido (PROCESSING, INVOICED, SHIPPED, DELIVERED, SHIPMENT_EXCEPTION) ?
- 1.10 Existe um fluxo natural de informações via API, envia-se o item, na sequencia estoque, preço e status. Os pedidos são consultados, e após aprovados e integrados, deverão ser informados os status PROCESSING, INVOICED, SHIPPED e DELIVERED. Quando existe a ocorrência de erro em uma das etapas mencionadas, as subseqüentes também são prejudicadas. Qual o tratamento dado para o item ou atualização de status de pedido quando um erro ocorre?
 - 1.10.1 Existe uma forma de retirar este produto ou atualização de pedido do fluxo para evitar um número de erros ainda maior? Qual?

5. PERGUNTAS FREQUENTES

Em Sandbox, ao enviar uma carga de produto, este será aprovado automaticamente?

As regras de aprovação para produto/sku devem ser consultadas em cada Marketplace. Pois cada um utiliza uma forma de validar os produtos/sku enviados para liberar os mesmos nos sites.

Em sandbox as informações só podem ser trocadas entre ERP e Integra, nada será enviado para o Marketplace. A comunicação com Marketplace é feita apenas em ambiente de produção.

Se eu enviar um PUT Product para reativar um produto, seus skus serão todos reativados automaticamente? Ou devo enviar reativações separadas para cada.

Existe ativação/inativação de produtos?

Existe o campo de Status para Produto, mas no momento este não é utilizado.

Futuramente criaremos a função para que a inativação de produto inative os Skus filhos, e a reativação funcione da mesma maneira.

Posso inativar apenas um sku de um product?

Sim é possível inativar um ou mais skus de um mesmo produto.

Para apenas inativar ou reativar posso mandar o JSON de put do product ou sku apenas com a propriedade "Status"? Exemplo:

PUT.... /api/Sku

```
{  
  "Status": false  
}
```

Não. É necessário passar o JSON inteiro com o Status false ou true.

Para alterar o preço de um sku, posso fazê-lo tanto pela api Price ou posso fazê-lo pela API de Put Sku?

Sim você pode fazer a alteração de preço tanto pela price, quanto pelo put sku. Mas é melhor fazer pelo PUT de price, para não sobrecarregar os servidores com dados a mais do que uma atualização necessita. O mesmo vale para o estoque.

Posso enviar alterações de Products/Sku a qualquer momento? Há alguma restrição?

Sim pode enviar alterações de produtos e skus a qualquer momento, lembrando que após um produto já estar à venda no site de um Marketplace, as informações atualizadas serão sempre preço e estoque. Em caso de alteração de cadastro como nome, categoria do produto, descrição, peso etc. Deve ser verificado a política de atualização de produto/ sku para cada marketplace específico.

No Magazine Luiza no entanto, não existe restrição, qualquer dado alterado do produto é reenviado, mesmo que o produto ou sku já esteja vendendo no site.

Quais os tamanhos e formatos que podem ser enviados nos skus?

Os campos de Atributo são do tipo string, podendo ser enviados qualquer tipo de tamanho ou formato. Verificar em cada Marketplace se há restrições no tamanho.

Qual o tamanho dos atributos das APIs, especialmente produtos e skus?

No Integra o tamanho é livre, mas é preciso verificar em também em cada Marketplace o tamanho que cada um aceita.

Quando envio um POST de criação de categoria, os dados enviados são das categorias em minha loja?

Sim, siga a estrutura de categorias de sua loja.

Os IDs enviados para cadastro de categorias serão os mesmos recuperados depois pelo GET de categorias?

Sim, os Ids enviados serão os mesmos recuperados.

O que significam os objetos 'CategoryMarketplace' no GET de categorias?

O Objeto CategoriesMarketplace serve para o caso de uma categoria no Integra ter um link com a categoria do Marketplace. No caso do Magazine Luiza isso não existirá e no get de categoria este objeto virá sempre como null.

Como funciona o processo de recuperação de pedidos (API GET Order) exatamente?

O processo de recuperação de pedidos vai funcionar na OrderQueue. Na fila OrderQueue haverá os pedidos com status New(Novo pedido do Marketplace ainda não pago) e Approved(Pedidos já aprovados). Os novos pedidos devem ser pesquisados pelo GET passando como parâmetro o status do pedido. Caso não vá trabalhar com os pedidos ainda não pagos, passe como parâmetro sempre o status Approved.

Pedidos com Status Approved vem com a data de aprovação. Informação não contida nos pedidos com status New.

Após GET na OrderQueue, faça um PUT nesta fila com o Id da OrderQueue, para que o pedido saia desta fila e não apareça nas próximas consultas GET.

Altere os status dos pedidos baixados da OrderQueue, por exemplo ao baixar o pedido como APPROVED, altere o status para PROCESSING. Isso fará com que na próxima consulta pela busca de pedidos APPROVED o mesmo não apareça.

Como faço para a cada chamada não retornar os mesmos pedidos que eu já baixei para minha plataforma?

Depois de tratados os pedidos basta fazer um PUT neles e alterar os Status. Assim quando consultar os novos pedidos com Status Approved ou New, você não baixa os pedidos já consultados anteriormente.

No GET Order, o que significa o campo StoreName?

StoreName vai vir sempre como MAGAZINELUIZA para o caso de pedidos provenientes do Magazine Luiza.

Como retorna que o pedido é do Magazine Luiza? A propriedade "MarketPlaceName" terá o valor "Magazine Luiza"?

A propriedade MarketPlaceName virá com o valor "magazineluiza"

TelephoneMainNumber e TelephoneSecondaryNumber => O telefone fixo e o celular virão em algum destes campos em específico, ou precisarei identificar quais serão estes?

Não vem discriminado telefone celular ou residencial. Por exemplo, poder vir o celular no MainNumber ou no SecondaryNumber, dependendo do cadastro que o cliente faz no site.

CustomerPjCnpj e CustomerPfCpf => Há casos em que os dois campos serão retornados com valor? E qual é o principal do cliente para inserirmos no pedido, o CNPJ ou CPF?

Se no cadastro no site o cliente se declara como pessoa Jurídica o campo CustomerPjCnpj vem preenchido, caso ele se cadastre como pessoa física o campo CustomerPfCpf vem preenchido.

Não há possibilidade dele se cadastrar como pessoa física e jurídica em uma mesma conta.

ReceiverName => Este campo possui sempre o mesmo valor de CustomerPfName ou CustomerPjCorporatename?

Este campo é referente ao destinatário da encomenda. Se algum cliente faz uma compra e escolhe que a entrega seja feita em outro endereço, para um destinatário diferente do comprador, este campo vem preenchido com o nome do recebedor.

InsertedErp => Onde este campo é utilizado?

Este campo é utilizado no momento em que você insere o pedido do Integra no seu ERP. Neste momento você pode fazer a alteração deste campo para true.

Os campos de "Invoiced" e "Shipped" serão preenchidos somente após nós enviarmos a atualização de status correspondente? E se no caso possuírmos pedidos com mais de 1 Nota Fiscal, como enviaríamos estes na integração de Invoiced?

Os Status Invoiced e Shipped precisam ser informados com os dados de nota fiscal e de despacho respectivamente.

Um mesmo pedido não pode ter mais de uma nota, o que é possível é uma nota conter os vários produtos vendidos no pedido.

UpdatedMarketplaceStatus e OrderStatus => Qual a diferença entre estes dois campos?

O campo **UpdatedMarketplaceStatus** é do tipo bool e o campo **OrderStatus** é do tipo string.

UpdatedMarketplaceStatus quando true, significa que o status do pedido no Marketplace foi efetivamente atualizado.

OrderStatus é referente ao status em que o pedido se encontra.

O "Total do Frete" do Order (TotalFreight) é a somatória dos campos de frete dos Products (Freight)? O "Desconto Total" do Order (TotalDiscount) também segue a mesma linha do frete, somando todos os valores do desconto por produto (Products: Discount)?

Esta informação está disponível na documentação. No item 2.8 em Considerações sobre os campos "Discount", "Freight" e "Price"

Na "simulação de frete" retornada pela API, se tiver mais de 1 produto na requisição, o frete de cada item (Items) será o rateio respectivo para o próprio?

Ex: Em uma Simulação de Frete com 2 Itens, onde o frete total do pedido será R\$ 10.00, o json de resposta com os fretes seguirá o formato abaixo (2 itens com o frete, neste caso, dividido entre eles):

```
{
  "packages": [
    {
      "items": [
        {
          "sku": "123"
          "quantity": 1
        },
        {
          "sku": "321",
          "quantity": 1
        }
      ],
      "delivery_options": [
        {
          "id": "1"
          "type": "conventional",
          "name": "Entrega normal",
          "price": "5.00",
          "delivery_days": 14,
        },
        {
          "id": "2"
          "type": "conventional",
          "name": "Entrega normal",
          "price": "5.00",
          "delivery_days": 14,
        }
      ]
    }
  ]
}
```

O frete será calculado separadamente para cada produto distinto na order.

No momento da venda ao requisitar o frete em nossa API Integra o frete será retornado da seguinte maneira.

Corpo do Post

```

1 {
2   "session_id": "132121321",
3   "zipcode": "37517000",
4   "items": [
5     {
6       "sku": "2010727",
7       "quantity": 2
8     },
9     {
10      "sku": "2010719",
11      "quantity": 30
12    }
13  ]
14 }

```

Resposta

```

1 {
2   "packages": [
3     {
4       "items": [
5         {
6           "sku": "2010727",
7           "quantity": 2
8         }
9       ],
10      "delivery_options": [
11        {
12          "id": "13",
13          "type": "conventional",
14          "name": "JadLog",
15          "price": "57.28",
16          "delivery_days": 24
17        }
18      ]
19    },
20    {
21      "items": [
22        {
23          "sku": "2010719",
24          "quantity": 30
25        }
26      ],
27      "delivery_options": [
28        {
29          "id": "13",
30          "type": "conventional",
31          "name": "JadLog",
32          "price": "63.77",
33          "delivery_days": 24
34        }
35      ]
36    }
37  ]
38 }

```

O valor do frete não será rateado.

Em GET Orders, no array de "Products", o valor do "Price" e "Freight" retornados serão sempre os mesmo enviados anteriormente na "Simulação de Frete"?

Price: Este campo é o valor líquido do produto, ou seja, o valor bruto menos o valor de descontos apresentados no campo Discount

Freight: Valor do frete para o item.

O campo "TotalAmount" será retornado com os valores do "Total Freight" e "Total Discount" já discriminados?

O campo TotalAmount é igual ao total dos produtos menos os descontos mais o valor do frete.

$\text{TotalAmount} = \text{Price}(\text{Valor bruto do produto} - \text{Discount}(\text{do item})) + \text{Freight}$

Qual é o Id real do order do Integracommerce? "IdOrder" ou "IdOrderMarketplace"? E qual a diferença entre estes 2 campos?

Utilize sempre o IdOrder para trabalhar com os pedidos. O IdOrderMarketplace vai ser igual o IdOrder, mas isso depende muito de Marketplace para Marketplace. Então trabalhe sempre com o IdOrder para Get Put e Post