

# eBay API 常见问题 & GraphQL API 最佳实践

## 免责声明:

本文内容用于解释eBay平台的各项操作流程或平台政策、规则。本文中的任何（包括隐含的）观点、建议、结论等仅供参考，不构成对您的最终操作的建议。相关政策、规则的正式版本以各平台独立公布的为准。eBay不就本文中的内容等对您的最终操作等做出任何保证或者承诺，您据此做出的任何决策与eBay公司及编制人员无关。

© 2026 eBay. All rights reserved. Confidential and proprietary.

# 内容概览

- 01 Marketplace User Account Deletion 市场用户账户删除
- 02 无法找到刊登、订单、退货等
- 03 Trading API / GetOrders 接口中的买家电话
- 04 Trading API 中的 SKU & 变体配置
- 05 多段运输 & eBay国际运送计划
- 06 Media API & eBay 图片服务
- 07 Markdown Promotion 降价促销
- 08 Business Policies 业务政策
- 09 eBay API 的访问问题

接入 eBay API 必须根据程序的具体情况选择其中之一

## A. 申请豁免，因业务原因不存储任何 eBay 数据

\*例如，对接开发过程中不存储任何相关数据。

## B. 订阅 Marketplace Account Deletion 通知

# 市场用户账户删除

 **Event Notification Delivery Method**

☐ Platform Notifications (push) ☒ Marketplace Account Deletion

Exempted from Marketplace Account Deletion ☒

**Exemption reason \***

☒ I do not persist eBay data

☐ I am an eBay vendor and handle data as per my contractual/legal agreement with eBay

 **Note:** Failure to provide correct information may result in penalties or having your account disabled.

**Additional information**

**Submit**

# 市场用户账户删除

接入 eBay API 必须根据程序的具体情况选择其中之一

A. 申请豁免，因业务原因不存储任何 eBay 数据

例如，对接开发过程中不存储任何相关数据。

**B. 订阅推送通知，并删除对应的eBay数据**

1. 按照文档配置终结点以准备响应挑战
2. 终结点完成响应挑战以订阅topic
3. 对来自eBay的推送响应HTTP 200, 201, 202, 204
4. 根据推送内容采取对应措施
5. 响应失败会重试，失败超过24小时会标记终结点失效

 **Event Notification Delivery Method**

☐ Platform Notifications (push) ☒ Marketplace Account Deletion

Exempted from Marketplace Account Deletion ☐

Email to notify if marketplace account deletion notification endpoint is down ⓘ  
 [Edit](#)

Marketplace account deletion notification endpoint ⓘ

Verification token \* ⓘ

[Send Test Notification](#) [Save](#)

# 无法找到刊登、订单、退货等

Trading API / GetOrders返回了空订单数组 🖱️

Fulfillment API / getOrder返回了无效订单id 🖱️

```
{
  "errors": [
    {
      "errorId": 32100,
      "domain": "API_FULFILLMENT",
      "category": "REQUEST",
      "message": "Invalid Order Id: '02-12345-67890'",
      "parameters": [
        {
          "name": "orderId",
          "value": "02-12345-67890"
        }
      ]
    }
  ]
}
```

```
<soapenv:Envelope xmlns:SOAP
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <GetOrdersResponse xmlns="urn:ebay:apis:eBLBaseComponents">
      <Timestamp>2026-09-03T00:00:11.456Z</Timestamp>
      <Ack>Success</Ack>
      <Version>1455</Version>
      <Build>E1289_CORE_APIXO_19220561_R1</Build>
      <PaginationResult>
        <TotalNumberOfPages>0</TotalNumberOfPages>
        <TotalNumberOfEntries>0</TotalNumberOfEntries>
      </PaginationResult>
      <HasMoreOrders>false</HasMoreOrders>
      <OrderArray/>
      <OrdersPerPage>100</OrdersPerPage>
      <PageNumber>1</PageNumber>
      <ReturnedOrderCountActual>0</ReturnedOrderCountActual>
    </GetOrdersResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

# 无法找到刊登、订单、退货等

先确认当前的 user token 是哪个卖家的

Trading API (XML/SOAP) / GetUser

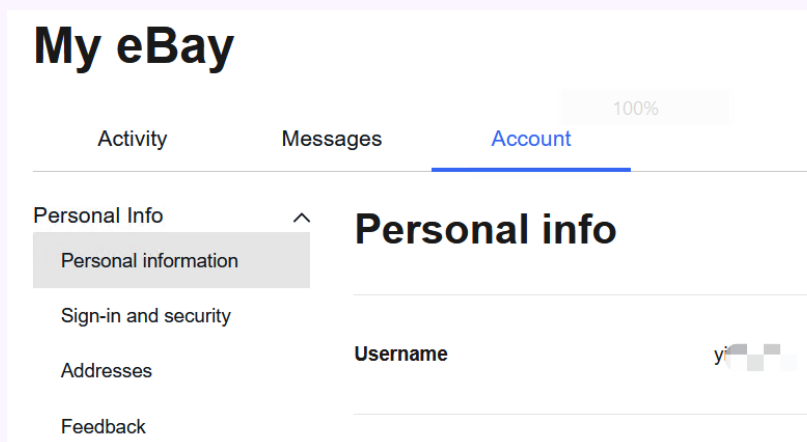
Identity API (Restful) / GetUser

User API (GraphQL) / user

对照eBay网页端的

<https://accountsettings.ebay.com/profile>

```
<?xml version="1.0" encoding="UTF-8"?>
<GetUserResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  ...
  <User>
    ...
    <RegistrationDate>2014-03-16T13:46:32.000Z</RegistrationDate>
    <Site>HK</Site>
    <Status>Confirmed</Status>
    <UserID>eBay username here</UserID>
    <UserIDChanged>>false</UserIDChanged>
    ...
  </User>
</GetUserResponse>
```



# 无法找到刊登、订单、退货等

先确认当前的 user token是哪个卖家的

Trading API (XML/SOAP) / GetUser

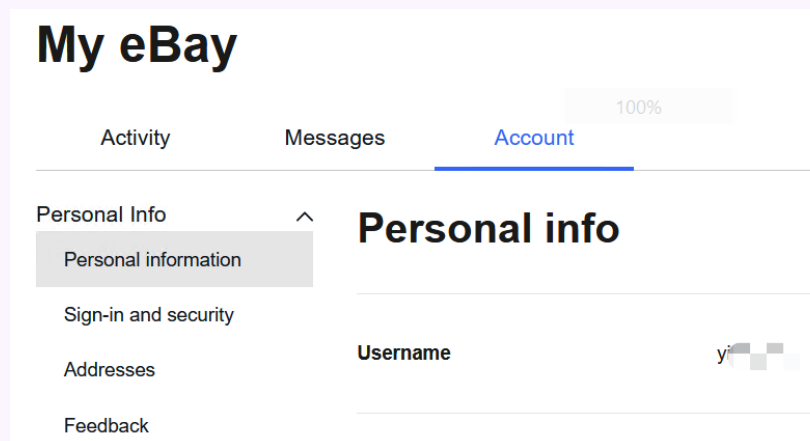
Identity API (Restful) / GetUser

User API (GraphQL) / user

```
{  
  ...  
  "accountType": "INDIVIDUAL",  
  "registrationMarketplaceId": "EBAY_HK",  
  "userId": "*****",  
  "username": "eBay username here"  
  ...  
}
```

对照eBay网页端的

<https://accountsettings.ebay.com/profile>



# 无法找到刊登、订单、退货等

先确认当前的 user token是哪个卖家的

Trading API (XML/SOAP) / GetUser

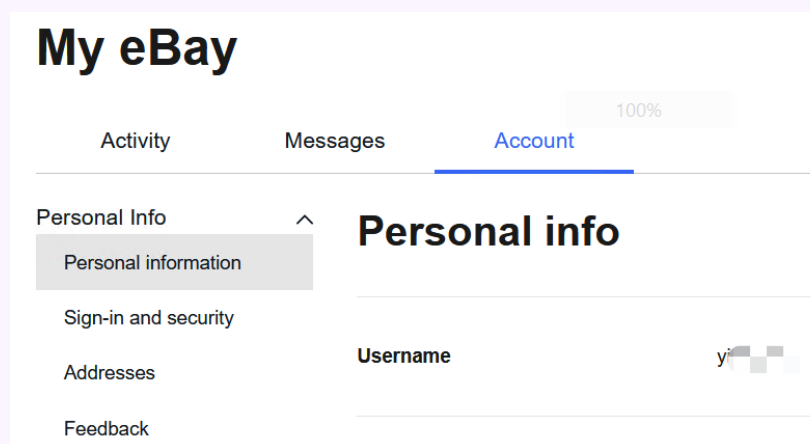
Identity API (Restful) / GetUser

User API (GraphQL) / user

对照eBay网页端的

<https://accountsettings.ebay.com/profile>

```
{
  "data": {
    "user": {
      "registrationMarketPlace": "EBAY_HK",
      "userCreatedAt": "2014-03-16T13:46:32.000Z",
      "userId": "*****",
      "userIdChanged": false,
      "username": "eBay username here",
    }
  }
}
```



# 无法找到刊登、订单、退货等

通过item id在网页端查看卖家的eBay用户名来对应API响应中的用户名

<https://www.ebay.com/itm/100012345678>

## PAIR LEFT RIGHT CLEAR HEADLIGHT LENS SHELL LAMPSHADE FOR SKODA OCTAVIA MK3 14-17



**PARTS LTD** (126436)

99.1% positive · Seller's other items ·

Message



**GBP 154.99**

Approximately US \$209.53

Condition: **New**

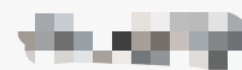
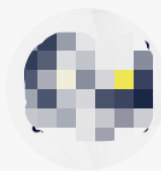
Compatibility: [See compatible vehicles](#)

Quantity:  **Last one** · 1 sold

Buy It Now

Add to cart

♥ Add to Watchlist



**PARTS LTD**

99.1% positive feedback 593K items sold 22K followers

Categories

Shop

Sale

**About**

Feedback

## About us

We are a automobile parts company that deals in top quality automobile parts and pieces.

Our main aim is customer satisfaction at affordable prices.

We always strive to make our clients comeback for more.

Location:

Member since:

Seller: **partsltd**

# TAPI / GetOrders 接口的买家电话

Invalid request – Trading API服务的问题，可以重试或尝试Fulfillment API / GetOrder

缺少国家区号 – 国家区号并不会返回，可以通过买家收货地址国家来对应

缺少电话号码 – 线下自提、线下合作店铺自提的订单

```
<ShippingAddress>
  <Name>Yinsheng Chen</Name>
  <Street1>Malden Road</Street1>
  <Street2>*****</Street2>
  <CityName>Liverpool</CityName>
  <StateOrProvince>Merseyside</StateOrProvince>
  <Country>GB</Country>
  <CountryName>United Kingdom</CountryName>
  <Phone>Invalid Request</Phone>
  <PostalCode>L6 6BE</PostalCode>
  <AddressId>100000000000000</AddressId>
  <AddressOwner>eBay</AddressOwner>
</ShippingAddress>
```

# TAPI / GetOrders 接口的买家电话

Invalid request – Trading API服务的问题，可以重试或尝试Fulfillment API / GetOrder

缺少国家区号 – 国家区号并不会返回，可以通过买家收货地址国家来对应

缺少电话号码 – 线下自提、线下合作店铺自提的订单

```
<ShippingAddress>  
  <Name>Yinsheng Chen</Name>  
  <Street1>Malden Road</Street1>  
  <Street2>*****</Street2>  
  <CityName>Liverpool</CityName>  
  <StateOrProvince>Merseyside</StateOrProvince>  
  <Country>GB</Country>  
  <CountryName>United Kingdom</CountryName>  
  <Phone>7411234567</Phone>  
  <PostalCode>L6 6BE</PostalCode>  
  <AddressId>100000000000000</AddressId>  
  <AddressOwner>eBay</AddressOwner>  
</ShippingAddress>
```

英国国家区号: **+44**

完整电话号码: **+44 7411234567**

# TAPI / GetOrders 接口的买家电话

Invalid request – Trading API服务的问题，可以重试或尝试Fulfillment API / GetOrder

缺少国家区号 – 国家区号并不会返回，可以通过买家收货地址国家来对应

缺少电话号码 – 线下自提、线下合作店铺自提的订单

```
<ShippingAddress>
  <Name>Yinsheng Chen</Name>
  <Street1> Srihari Super Store</Street1>
  <Street2> eCP:RAYUKYRT 94 Breck Road Anfield</Street2>
  <CityName>LIVERPOOL</CityName>
  <StateOrProvince/>
  <Country>GB</Country>
  <CountryName>United Kingdom</CountryName>
  <Phone/>
  <PostalCode>L4 2RD</PostalCode>
  <AddressOwner>eBay</AddressOwner>
  <AddressAttribute
type="ReferenceNumber">eCP:MPZWXVUL</AddressAttribute>
</ShippingAddress>
```

```
<PickupMethodSelected>
  <PickupMethod>PickUpDropOff</PickupMethod>
  <PickupStoreID>*****</PickupStoreID>
  <PickupStatus>PendingMerchantConfirmation</PickupStatus>
  <PickupFulfillmentTime>2026-09-
07T23:00:00.000Z</PickupFulfillmentTime>
  <PickupLocationUUID>*****</PickupLocationUUID>
</PickupMethodSelected>
```

# TAPI / SKU & 变体配置

## Stock Keeping Unit, 库存单位

商家自定的商品标识符；如使用变体则变体都需要独立的SKU；也可以是代替ApplicationData的自定义字段附加业务标记。

因刊登级别的SKU和某变体中的SKU值相同，导致修改操作报错。InventoryTrackingMethod不可以为SKU，否则无法修改SKU值。

SKU大小写敏感，字符不完全一致导致找不到对应刊登或变体报错。

# TAPI / SKU & 变体配置

## Stock Keeping Unit, 库存单位

商家自定的商品标识符；如使用变体则变体都需要独立的SKU；也可以是代替ApplicationData的自定义字段附加业务标记。

因刊登级别的SKU和某变体中的SKU值相同，导致修改操作报错。InventoryTrackingMethod不可以为SKU，否则无法修改SKU值。

SKU大小写敏感，字符不完全一致导致找不到对应刊登或变体报错。

```
<?xml version="1.0" encoding="UTF-8"?>
<RevisInventoryStatusResponse
xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2026-09-07T07:37:28.445Z</Timestamp>
  <Ack>Failure</Ack>
  <Errors>
    <ShortMessage>Cannot revise a Multi-SKU item when item
level SKU is supplied.</ShortMessage>
    <LongMessage>Variation level SKU or Variation level SKU and
ItemID should be supplied to revise a Multi-SKU
item.</LongMessage>
    <ErrorCode>21916735</ErrorCode>
    <SeverityCode>Error</SeverityCode>
    <ErrorParameters ParamID="ItemID">
      <Value>100000010001</Value>
    </ErrorParameters>
    <ErrorParameters ParamID="SKU">
      <Value>ITEM-SKU</Value>
    </ErrorParameters>
    <ErrorClassification>RequestError</ErrorClassification>
  </Errors>
  <Version>1457</Version>
  <Build>1457_UNI_API5_19110934_R1</Build>
</RevisInventoryStatusResponse>
```

# TAPI / SKU & 变体配置

## Stock Keeping Unit, 库存单位

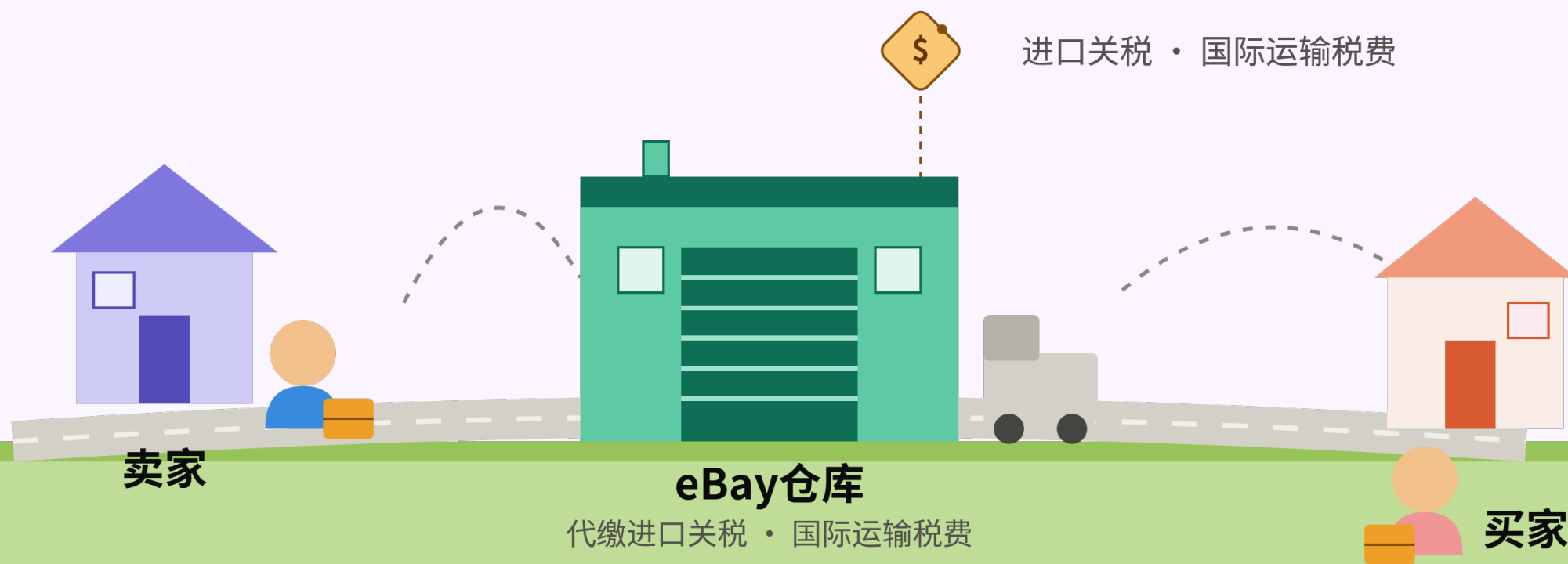
商家自定的商品标识符；如使用变体则变体都需要独立的SKU；也可以是代替ApplicationData的自定义字段附加业务标记。

因刊登级别的SKU和某变体中的SKU值相同，导致修改操作报错。InventoryTrackingMethod不可以为SKU，否则无法修改SKU值。

SKU大小写敏感，字符不完全一致导致找不到对应刊登或变体报错。

```
<?xml version="1.0" encoding="UTF-8"?>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2026-09-07T07:37:28.445Z</Timestamp>
  <Ack>Failure</Ack>
  <Errors>
    <ShortMessage>No Matching SKU</ShortMessage>
    <LongMessage>There is no active item matching the specified
SKU</LongMessage>
    <ErrorCode>21916270</ErrorCode>
    <SeverityCode>Error</SeverityCode>
    <ErrorClassification>RequestError</ErrorClassification>
  </Errors>
  <Version>1457</Version>
  <Build>1457_UNI_API5_19110934_R1</Build>
</GetItemResponse>
```

# 多段运输 & eBay国际运送计划



# 多段运输 & eBay国际运输计划

Trading API / GetOrders 中参考：

- **ShippingServiceSelected**显示eBay国际运输
- **isMultiLegShipping**判定是否有多段运输
- **actualShippingCost**是买家付的运费总额（卖家默认的运输费 + eBay国际运输的费用）
- **MultiLegShippingDetails.SellerShipmentToLogisticsProvider.ShippingServiceDetails.TotalShippingCost**为卖家到eBay仓的运费
- **MonetaryDetails**通常包含分别付给卖家和ebay的费用，付给ebay的费用包含运输税费

```
...
<ShippingServiceSelected>
  <ShippingService>eBayIntermidatedInternationalPriorityShippingDDP</ShippingService>
  <ShippingServiceCost
    currencyID="USD">80.00</ShippingServiceCost>
    <ImportCharge currencyID="USD">100.0</ImportCharge>
    <ShippingPackageInfo>
      <ActualDeliveryTime>2026-09-
07T09:00:00.000Z</ActualDeliveryTime>
    </ShippingPackageInfo>
  </ShippingServiceSelected>
...
<IsMultiLegShipping>true</IsMultiLegShipping>
...
```

# 多段运输 & eBay国际运输计划

## Trading API / GetOrders 中参考：

- ShippingServiceSelected显示eBay国际运输
- isMultiLegShipping判定是否有多段运输
- actualShippingCost是买家付的运费总额（卖家默认的运输费 + eBay国际运输的费用）
- MultiLegShippingDetails.SellerShipmentToLogisticsProvider.ShippingServiceDetails.TotalShippingCost为卖家到eBay仓的运费
- MonetaryDetails通常包含分别付给卖家和ebay的费用，付给ebay的费用包含运输税费

```
...
<TransactionArray>
  <Transaction>
    ...
    <ActualShippingCost currencyID="USD">112.52
  </ActualShippingCost>
    ...
  </Transaction>
</TransactionArray>
...
<MultiLegShippingDetails>
  <SellerShipmentToLogisticsProvider>
    <ShippingServiceDetails>
      <ShippingService>ShippingMethodStandard</ShippingService>
      <TotalShippingCost currencyID="USD">0.0
    </TotalShippingCost>
    </ShippingServiceDetails>
    ...
  </SellerShipmentToLogisticsProvider>
</MultiLegShippingDetails>
```

# 多段运输 & eBay国际运输计划

## Trading API / GetOrders 中参考:

- ShippingServiceSelected显示eBay国际运输
- isMultiLegShipping判定是否有多段运输
- actualShippingCost是买家付的运费总额  
(卖家默认的运输费 + eBay国际运输的费用)
- MultiLegShippingDetails.SellerShipmentToLogisticsProvider.ShippingServiceDetails.TotalShippingCost为卖家到eBay仓的运费
- MonetaryDetails通常包含分别付给卖家和eBay的费用, 付给eBay的费用包含运输税费

```
<MonetaryDetails>
  <Payments>
    <Payment>
      <PaymentStatus>Succeeded</PaymentStatus>
      <Payer type="eBayUser">the_buyer</Payer>
      <Payee type="eBayUser">the_seller</Payee>
      <PaymentTime>2026-09-07T06:00:00.000Z</PaymentTime>
      <PaymentAmount currencyID="USD">1500.0</PaymentAmount>
      <ReferenceID
type="ExternalTransactionID">10000000000000</ReferenceID>
      <FeeOrCreditAmount currencyID="USD">0.0</FeeOrCreditAmount>
    </Payment>
    <Payment>
      <PaymentStatus>Succeeded</PaymentStatus>
      <Payer type="eBayUser">the_buyer</Payer>
      <Payee type="eBayUser">eBay</Payee>
      <PaymentTime>2026-09-07T06:00:00.000Z</PaymentTime>
      <PaymentAmount currencyID="USD">180.0</PaymentAmount>
      <ReferenceID
type="ExternalTransactionID">10000000000001</ReferenceID>
    </Payment>
  </Payments>
</MonetaryDetails>
```

# Media API & eBay 图片服务

## 内容限制

- 文件过大／时长过长
- 画质不佳
- 违法内容
- 冒犯性内容
- 裸露或性相关内容
- 侵权内容／未获授权素材
- 导流站外交易

参考: <https://www.ebay.com/help/policies/listing-policies/images-text-policy?id=4240>

# Media API & eBay 图片服务

## 功能限制

新上传的图片 / 视频有**30天**有效期，未使用到刊登的内容默认30天自动过期，使用在刊登上会自动续期。

getImage和getVideo响应返回404的时候说明资源已被删除，CDN缓存可以访问也不可再使用。

```
{  
  "errors": [  
    {  
      "errorId": 190200,  
      "domain": "API_MEDIA",  
      "category": "REQUEST",  
      "message": "The image resource was not found."  
    }  
  ]  
}
```



# Media API & eBay 图片服务

## 访问限制

eBay会在上传时，访问开发者提供的URL来下载资源。

为预防盗链，可以参考使用预签名链接，例如：

**`https://img.abc.com/listingpicture001.png?signed=1234567890`**

# Markdown Promotion 降价促销

- `getListingsSet` 接口的 `listingPromotionStatuses` 字段中的状态只有降价促销本身发生变化时才会进行更新

|             |              |
|-------------|--------------|
| PENDING     | 等待处理         |
| MARKED_DOWN | 降价成功         |
| MARKED_UP   | 已不在降价促销,恢复原价 |
| SKIPPED     | 跳过未处理        |
| FAILED      | 失败,检查配置      |

- 降价促销导致价格变化会在rvh页面留下记录
- 获取准确的状态可以使用 `GetItem`
- 降价促销结束之后至少留出3小时再开始下一个降价促销

```
{
  "listingId": "365000000000",
  "inventoryReferenceId": "SMR12345",
  "title": "10x Self-made Rings",
  "currentPrice": {
    "value": "188.00",
    "currency": "USD"
  },
  "listingCondition": "NEW",
  "quantity": 10,
  "freeShipping": true,
  "listingCategoryId": "171512",
  "storeCategoryId": "1",
  "listingConditionId": "1000",
  "inventoryReferenceType": "INVENTORY_ITEM",
  "listingPromotionStatuses": [
    {
      "statusChangedDate": "2026-09-07T01:00:00.000Z",
      "listingMarkdownStatus": "MARKED_DOWN",
      "statusMessage": "The item was successfully discounted."
    }
  ]
}
```

# Markdown Promotion 降价促销

- `getListingsSet` 接口中的 `listingPromotionStatuses` 字段中的状态只有降价促销本身发生变化的时候才会进行更新

|             |              |
|-------------|--------------|
| PENDING     | 等待处理         |
| MARKED_DOWN | 降价成功         |
| MARKED_UP   | 已不在降价促销,恢复原价 |
| SKIPPED     | 跳过未处理        |
| FAILED      | 失败,检查配置      |

|              |                   |                                                                                                                                   |
|--------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Aug 10, 2026 | 12:32:47 AM GMT+8 | <a href="#">Buy It Now Price</a><br><a href="#">Sale End</a><br><a href="#">Details</a><br><a href="#">Buy It Now Price for</a>   |
| Jun 30, 2026 | 04:46:54 PM GMT+8 | <a href="#">Buy It Now Price</a><br><a href="#">Sale Start</a><br><a href="#">Details</a><br><a href="#">Buy It Now Price for</a> |

- 降价促销导致价格变化会在rvh页面留下记录
- 获取准确的状态可以使用 `GetItem`
- 降价促销结束之后至少留出3小时再开始下一个降价促销

# Markdown Promotion 降价促销

- `getListingsSet` 接口中的 `listingPromotionStatuses` 字段中的状态只有降价促销本身发生变化的时候才会进行更新

|             |              |
|-------------|--------------|
| PENDING     | 等待处理         |
| MARKED_DOWN | 降价成功         |
| MARKED_UP   | 已不在降价促销,恢复原价 |
| SKIPPED     | 跳过未处理        |
| FAILED      | 失败,检查配置      |

- 降价促销导致价格变化会在rvh页面留下记录
- 获取准确的状态可以使用 `GetItem`
- 降价促销结束之后至少留出3小时再开始下一个降价促销

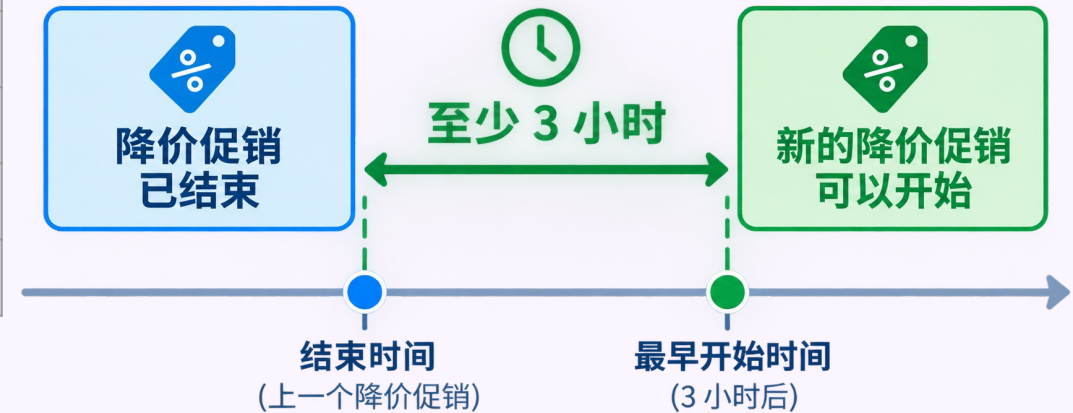
```
<?xml version="1.0" encoding="UTF-8"?>
<GetItemResponse xmlns="urn:ebay:apis:eBLBaseComponents">
  <Timestamp>2026-09-08T00:35:00.000Z</Timestamp>
  <Ack>Success</Ack>
  ...
  <SellingStatus>
    ...
    <PromotionalSaleDetails>
      <OriginalPrice currencyID="USD">288.00</OriginalPrice>
      <StartTime>2026-09-07T15:32:25.000Z</StartTime>
      <EndTime>2026-09-15T15:32:00.000Z</EndTime>
    </PromotionalSaleDetails>
    ...
  </SellingStatus>
  ...
  <Version>1457</Version>
  <Build>1457_UNI_API5_19110934_R1</Build>
</GetItemResponse>
```

# Markdown Promotion 降价促销

- getListingsSet 接口中的 listingPromotionStatuses 字段中的状态只有降价促销本身发生变化的时候才会进行更新

|             |              |
|-------------|--------------|
| PENDING     | 等待处理         |
| MARKED_DOWN | 降价成功         |
| MARKED_UP   | 已不在降价促销,恢复原价 |
| SKIPPED     | 跳过未处理        |
| FAILED      | 失败,检查配置      |

- 降价促销导致价格变化会在rvh页面留下记录
- 获取准确的状态可以使用GetItem
- 降价促销结束之后至少留出3小时再开始下一个降价促销



# Business Policies 业务政策

业务政策简化了重复和复杂的履约、退货、付款政策

```
<PaymentMethods>VALID_SITE_PAYMENT_METHOD</PaymentMethods>
<AutoPay>false</AutoPay>
<ShippingDetails>
  <ShippingType>Flat</ShippingType>
  <ShippingServiceOptions>
    <ShippingService>VALID_DOMESTIC_SHIPPING_SERVICE</ShippingService>
    <ShippingServiceCost
currencyID="USD">4.99</ShippingServiceCost>
    <ShippingServicePriority>1</ShippingServicePriority>
  </ShippingServiceOptions>
</ShippingDetails>
<ReturnPolicy>
  <ReturnsAcceptedOption>ReturnsAccepted</ReturnsAcceptedOption>
  <RefundOption>MoneyBack</RefundOption>
  <ReturnsWithinOption>Days_30</ReturnsWithinOption>
  <ShippingCostPaidByOption>Buyer</ShippingCostPaidByOption>
</ReturnPolicy>
```

```
<SellerProfiles>
  <SellerPaymentProfile>
    <PaymentProfileID>paymentPolicyId</PaymentProfileID>
    <PaymentProfileName>paymentPolicyName</PaymentProfileName>
  </SellerPaymentProfile>
  <SellerReturnProfile>
    <ReturnProfileID>returnPolicyId</ReturnProfileID>
    <ReturnProfileName>returnPolicyName</ReturnProfileName>
  </SellerReturnProfile>
  <SellerShippingProfile>
    <ShippingProfileID>fulfillmentPolicyId</ShippingProfileID>
    <ShippingProfileName>fulfillmentPolicyName</ShippingProfileName>
  </SellerShippingProfile>
</SellerProfiles>
```

# Business Policies 业务政策

## 判断卖家账户是否启用业务政策

通过 Account API v1 / getOptedInPrograms 判定。  
如果已启用，则需要使用政策。

```
{
  "programs": [
    {
      "programType": "OUT_OF_STOCK_CONTROL"
    },
    {
      "programType": "SELLING_POLICY_MANAGEMENT"
    }
  ]
}
```

## 业务政策已启用

应使用定义好的履约、退货、付款政策，如继续直接在刊登操作时手动定义则会自动创建新政策。

```
<Errors>
  <ShortMessage>Seller has opted into business policies.
Please use policy IDs rather than legacy fields for Shipping,
Payments, or Returns.</ShortMessage>
  <LongMessage>Seller has opted into business policies.
Please use policy IDs rather than legacy fields for Shipping,
Payments, or Returns, otherwise new policies may be
automatically created on the seller's behalf.</LongMessage>
  <ErrorCode>21919456</ErrorCode>
  <SeverityCode>Warning</SeverityCode>
  <ErrorClassification>RequestError</ErrorClassification>
</Errors>
```

# Business Policies 业务政策

## 接口返回异常

自动创建过多的政策会导致获取接口异常，且接口不支持分页。

GET /sell/account/v1/fulfillment\_policy?marketplace\_id=EBAY\_GB

```
{
  "errors": [
    {
      "errorId": 20500,
      "domain": "API_ACCOUNT",
      "category": "APPLICATION",
      "message": "System error."
    }
  ]
}
```

# eBay API 的访问问题

## 连接性问题

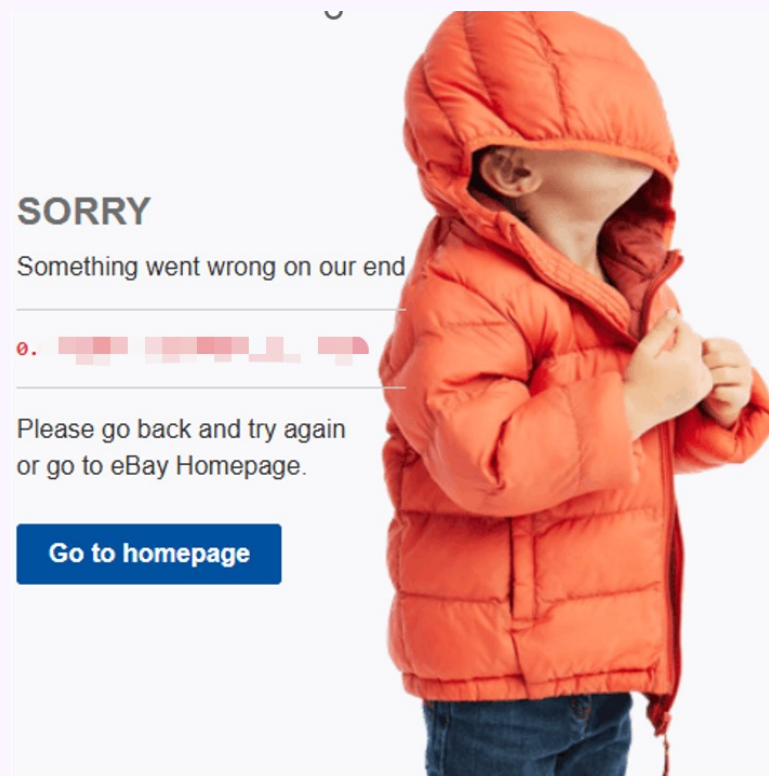
eBay对api.ebay.com终结点的接口有特别路线以提升可用性。

内地至国际的网络波动会出现调用超时的问题。

Connect to api.ebay.com:443  
[api.ebay.com/209.140.142.13] failed:  
connect timed out; nested exception is  
org.apache.http.conn.ConnectTimeoutExc  
eption: **Connect to api.ebay.com:443**  
**[api.ebay.com/209.140.142.13] failed:**  
**connect timed out**

## 非连接性问题

接口返回非结构性内容，反而是HTML错误页面。



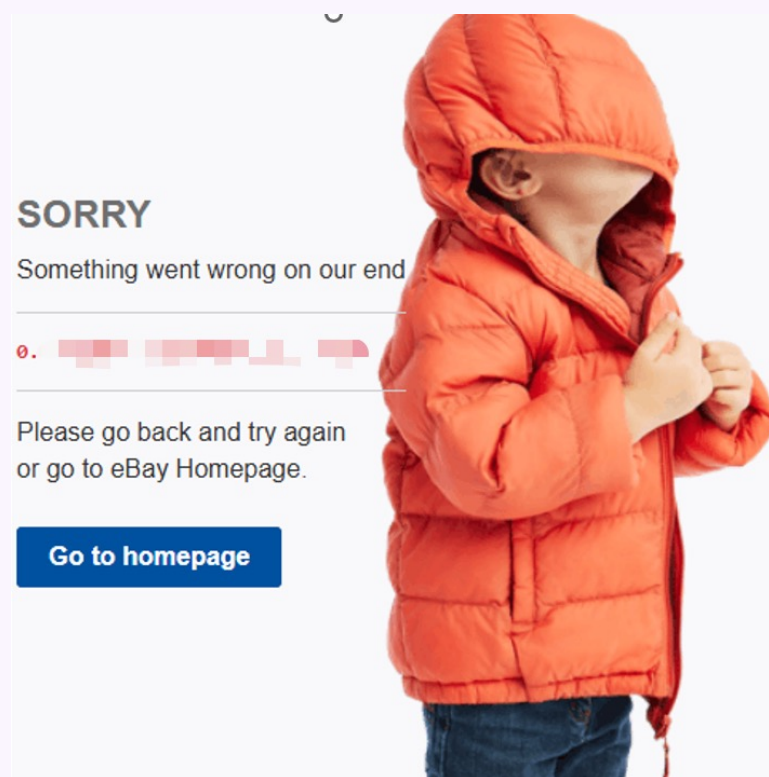
# eBay API 的访问问题

## 连接性问题

- 做重试逻辑，同步操作进队列处理。
- 适当延长超时时间，以保证在不稳定时服务器有更多和API服务器连接。
- 遵守国家法律法规的前提下，向运营商申请跨境网络专线和使用特定地区的服务器以提升稳定性。

## 非连接性问题

发邮件至**DL-eBay-CBT-API@ebay.com**咨询。



# eBay GraphQL 最佳实践

---

如何稳定且高效的使用eBay GraphQL API

为什么选择GraphQL

# 一次请求，拿到你刚好需要的数据

同一个商品详情需求

SOAP or REST  
GET /product/{id}  
GET /product/{id}/reviews  
两次调用 + 固定响应里可能有未使用字段

GraphQL  
query GetProduct(\$id: ID!) {  
 product(id: \$id) {  
 title  
 reviews {  
 id  
 rating  
 comment  
 }  
 }  
}  
一次请求，精确遍历图拿到所需数据

示例GraphQL API 响应

HTTP Status Code: 200

```
{  
  "data": {  
    "product": {  
      "title": "Headphone",  
      "reviews": [  
        {  
          "id": "r1",  
          "rating": 5,  
          "comment": "Great"  
        },  
        {  
          "id": "r2",  
          "rating": 4,  
          "comment": "Good"  
        }  
      ]  
    }  
  }  
}
```

为什么值得切换

| 对比维度  | SOAP/REST     | GraphQL      |
|-------|---------------|--------------|
| 字段范围  | 固定范围，常冗余      | 用户可以按需选择     |
| 相关资源  | 常需多API端点      | 一次遍历Graph    |
| 契约发现  | 分散在多个文档       | 统一的强类型schema |
| API演进 | 常按版本区分(v1/v2) | 无需版本号，向前兼容   |

注意：深嵌套、过多根字段、过多Alias字段的请求需要避免

基本遵循规范

## 一个URL，一套JSON + OAuth 2.0 契约

### REQUEST CHECKLIST

---

#### POST

query与mutation使用POST，不支持  
Get

---

#### 固定URL

<https://graphqlapi.ebay.com/graphql>

---

#### Application/json

request / response 一致

---



#### OAuth 2.0

Bearer token(必须) + marketplace (可选)

注意: 不支持 Content-Type: application/graphql-response+json

### 示例最简请求 - 完整的Transport Contract

```
curl --request POST 'https://graphqlapi.ebay.com/graphql' \  
--header 'Authorization: Bearer <access-token>' \  
--header 'X-EBAY-C-MARKETPLACE-ID: EBAY_US' \  
--header 'Content-Type: application/json' \  
  
--data '{  
  "query": "query MyListing($listingId: ID!) { sellerListings(input: { listings: {  
    listingId: $listingId } }) { ... on ListingsSuccess { __typename listings { ... on  
    ListingSuccess { __typename listingId } } } }",  
  "variables": {  
    "listingId": "123456"  
  }  
}'
```

使用JSON serializer; 不推荐手工拼接和转义 document

什么是好的 QUERY + MUTATION

## 使用命名Operation，显示Variable，分支处理Outcome



```
mutation ApproveMyReturn($returnId: ID!) {  
  approveReturn(input: { returnId: $returnId }) {  
    __typename  
    ... on ApproveReturnInputValidationError {  
      errorCode  
    }  
    ... on OrderReturn {  
      closedAt  
    }  
    ... on ReturnActionNotPermittedError {  
      errorCode  
    }  
    ... on ReturnNotFoundError {  
      errorCode  
    }  
  }  
}
```

```
const result = response.data?.approveReturn  
switch (result.__typename) {  
  case "OrderReturn":  
    showReturnClosed(result.closedAt)  
    break  
  case "ApproveReturnInputValidationError":  
    showError(result.errorCode)  
    break  
  case "ReturnActionNotPermittedError":  
    showError(result.errorCode)  
    break  
  case "ReturnNotFoundError":  
    showError(result.errorCode)  
    break  
  default:  
    showUnavailableState()  
}
```

Schema 演进：nullable字段按可空值处理；未知 enum 保守处理；新增 union / interface member 采用 fallback

如何处理分页

## 使用Cursor查询大集合数据

### 1. FIRST PAGE

```
searchInput: {
  pagination: {
    maxPageSize: $size,
  },
  filters: {
    formatType: AUCTION
  }
}
```

第一次请求不发送pageCursor

### 2. RESPONSE DATA

```
"activeListings": {
  "__typename": "SellerListings",
  "pagination": {
    "nextCursor": "eyJwYWdlIjoyfQ=="
  },
  "listings": [
    {"listingId": "L1234567890"},
    {"listingId": "L1234567891"}
  ]
}
```

Cursor是一种token，不是偏移量或页码

### 3. NEXT PAGE

```
searchInput: {
  pagination: {
    maxPageSize: $size,
    pageCursor: $nextCursor
  },
  filters: {
    formatType: AUCTION
  }}
}
```

在遍历分页时需保持 filters / sort 条件一致

``nextCursor = null`` → 停止

`cursor` 过期 → 从第一页重启

不要解析/修改`cursor` → 无效

Cursor navigation 不等于任意“第 X 页”跳转；第一页是否锁定 maxPageSize 以具体字段文档为准

如何处理不同的HTTP Status Code 以及 Error

## HTTP 200 不代表执行完全成功

| Response                            | 含义               | 用户端处理方式               |
|-------------------------------------|------------------|-----------------------|
| HTTP 500                            | 平台 / 上游失败        | 按文档或者官方通知重试           |
| HTTP 400                            | 缺失token; 请求文档不合法 | 修正后再进行重试              |
| HTTP 401                            | 无效的OAuth2 token  | 重新生成后重试               |
| HTTP 422                            | 请求复杂度超限          | 检查并更新请求后再重试           |
| HTTP 200 with data only             | 请求执行成功并且没有任何错误   | 正常处理                  |
| HTTP 200 with data and errors       | 请求执行部分成功         | 检查errors里包含的错误, 做相应处理 |
| HTTP 200 with data: null and errors | 请求执行失败           | 根据errorCode做相应处理      |

HTTP 200: errorCode + path 决定如何处理

```
{ // Bad input case
  "data": { "calculateDiscount": null },
  "errors": [{
    "message": "Discount percentage must be between 0
and 100",
    "path": ["calculateDiscount"],
    "extensions": {
      "errorCode": "BAD_INPUT_VALUE"
    }
  ]
}

{ // Rate limit case
  "data": null,
  "errors": [{
    "message": "Rate limit exceeded.",
    "extensions": {
      "errorCode": "RATE_LIMITED",
      "limitedResource": {
        "resourceType": "ROOT_OPERATION_FIELD",
        "name": "Query.sellerListings"
      }
    }
  ]
}
```

常见的 extensions.errorCode: FORBIDDEN, RATE\_LIMITED,  
BAD\_INPUT\_VALUE, INTERNAL\_SERVER\_ERROR

[https://developer.ebay.com/develop/api/sell/error\\_codes](https://developer.ebay.com/develop/api/sell/error_codes)

# 确保可用性与可预测性能

| 指标      | 阈值   | 用户端处理方式          |
|---------|------|------------------|
| 请求的深度   | 15   | 减少嵌套字段或拆分请求      |
| 请求字段总数  | 2000 | 删去不需要的字段         |
| 根字段总数   | 11   | 减少顶层方法(根字段)调用    |
| 别名根字段总数 | 11   | 使用支持批量输入的方法或拆分请求 |

## HTTP 422 – Unprocessable Query

```
{
  "errors": [
    {
      "extensions": { "errorCode": "MAX_COMPLEXITY_EXCEEDED" },
      "message": "Query complexity exceeds the maximum allowed. Your
query: depth=18; maximum allowed=15."
    }
  ]
}
```

请求深度：5

```
query {
  user {
    posts {
      comments {
        author {
          name
        }
      }
    }
  }
}
```

根字段总数：4

```
query {
  user { name }
  posts { count }
  friends { count }
  followers {count }
}
```

请求字段总数：6

```
query {
  user {
    id
    name
    posts {
      id
      title
    }
  }
}
```

别名根字段总数：4

```
query {
  user1: user(id:1) { name }
  user2: user(id:2) { name }
  user3: user(id:3) { name }
  user4: user(id:4) { name }
}
```

## 系统判定标准：

- 用户请求必须满足上述四项全部的标准
- 在更新请求达到标准前不要重试
- 将来可能还会引入Query Cost Limitation



# 谢谢

