

# Otter开放接口开发文档

该文档也可以在线查看，请向我们索取登陆账号与密码

<https://developer.tryotter.com/docs#>

最后更新:2022.01.10

- 总览 **Overview**
- 引导 **Guides**
- 接口 **Endpoints**
  - 验证 Auth
  - 菜单 Menus
  - 订单 Orders
  - 测试 Ping
  - 配送 Delivery
- 数据模型 - **MODELS**
  - 订单 Order
  - 菜单 Menus
  - 错误信息 Error Message
- 回调模式 **WEBHOOK MODELS**
  - 事件推送消息 Event Notification
  - 订单 Order
  - 订单历史状态 Order Status History

## 总览 Overview

水獭掌柜API接口是基于RESTful原则开发并同时使用OAuth2.0协议保护。除了提供的接口，API同时提供回调模式给贵公司的系统发送一系列的消息，从而实现双方系统的双向信息同步。我们也同时提供一个测试环境供您开发调试

The API endpoints are developed around RESTful principles secure via the OAuth2.0 protocol.

Beyond the entry points, the API also provides a line of communication into your system via webhooks.

For testing purposes, we offer a staging environment.

## 引导 Guides

我们目前正在搭建能够提供自助式服务的开发者平台，在这之前我们会提供相关咨询服务，所以请您务必与我们的团队保持紧密联系，确保您能及时获得所有的信息和帮助。

We are currently building a developer portal so most configurations can be self-serve, but until that is complete we are offering a white-glove experience, so please maintain close contact with your Account Representative for all your onboarding needs.

## 流程开始 Getting Started

### 申请开通 Onboard Your Application

外部开发者需要向我们发起一次性的申请创建账号，我们将为您的应用设置授权。请联系我们的团队为您注册。

External developers are required to have a registered Application representing their integration. This is a one time, manual registration needed to setup authentication and authorization between your application and the API. *Please reach out to your Account Representative to register your Application with the API.*

一旦注册完成，您将收到生产环境的Application ID以及Client Secret，请您妥善保管。同时我们将指引您如何使用这些授权。这些授权信息将由我们的团队发送给您。

注释：Application ID曾经使用名称Partner ID，所以如果您是在前几个版本开发，请及时更新。

Once onboarded, your Application will receive its own Application ID and Client Secret for the production environment. This Guide will walk you through where and how to use these credentials.

*NOTE: The Application ID used to be called the Partner ID, so please use those interchangeably.*

## 搭建回调模式 **Onboard Your Webhook Configurations**

我们的API可以支持您系统的回调模式，请联系我们的团队为您配置

The API allows you to configure your integration to receive Webhook into your system. *Please reach out to your Account Representative to register to receive Webhooks from the API.*

## 创建店铺 **Onboard Your Store(s)**

我们的系统只提供以店铺为颗粒度的集成方式，以便我们可以将外部店铺的信息和我们系统内部的门店信息进行匹配。在此处需要您开发一套CKDSS来存储所有的门店店铺匹配信息。具体请联系我们的团队为您提供更多信息。

Applications are required to register at every store to authorize requests at a store level. This registration is on a per-store basis, as to associate the store entity in your system with the store entity in our system. *Reach out to your Account representative to onboard each store that your application acts on behalf of.*

## 生成客户端**Token Generating Client Token**

请按照API Auth部分提供的具体描述来为您的系统生成Token。我们的token是有有效期的，所以您的系统一定需要具备自动更新token的功能。

Follow the description in the API Auth to generate a token from your secret.

Client tokens expire, so integrations must handle expired tokens pre-emptively and/or retroactively.

## 调用测试 **Test Invocation**

我们提供一个ping服务器的接口来测试与我们系统是否成功链接，代码如下We provide a ping endpoint that allows you to test the connectivity with our systems.

```
curl --location --request GET 'https://partners-staging.baitime.cn/v1/ping' \  
--header 'Authorization: Bearer [CLIENT_CREDENTIALS_TOKEN]' --header \  
'X-Application-Id: [APPLICATION_ID]'
```

如果返回的状态是200，那么证明您的授权有效，可以开始下一步的开发。

如果返回的是401，那么证明您的授权已经失效，需要重新生成。

如果返回的是5xx，那么证明我们的系统有问题，请及时联系我们的团队。

在下面的Ping接口中可以找到更多相关信息。

If the status result of this invocation is a 200, then your auth parameters are valid and ready to go!

If the status result of this invocation is a 401, then your token has EXPIRED and should be regenerated.

If the status result of this invocation is a 5xx, then something went wrong in our system.

Read more about the Ping Endpoint in the documentation.

## API

我们的API通过JSON与源URL通信，同时综合使用HTTP headers、状态代码以及HTTP的不同方法（如Get/Post等）执行对应的具体业务逻辑。

举例：如下是一个从我们接口利用GET调用数据的代码

The APIs use resource-oriented URLs communicating, primarily, via JSON and leveraging the HTTP headers, response status codes, and verbs.

To exemplify how the API is to be consumed, consider a fake GET resource endpoint invocation below:

```
curl --request GET 'https://partners-staging.baitime.cn/v1/resource/123' \  
--header 'Authorization: Bearer 34fdabeeafds=' --header 'X-Store-Id: 321' --header \  
'X-Application-Id: e22f94b3-967c-4e26-bf39-9e364066b68b'
```

Header	描述 Description
Authorization	标准的HTTP header会包含发起请求方的授权信息。header的内容是一个 Bearer 开头的授权token ( 在下一步详细解释如何申请 ) Standard HTTP header is used to associate the request with the originating invoker. The content of this header is a Bearer token generated from client_secret, defined in the API Auth guide.
X-Application-Id	在上线时给您提供的纯文本格式的application ID The plain-text Application Id, provided at onboarding.
X-Store-Id	您系统中存储的店铺ID The ID of the store in your system this call acts on behalf of.

所有的接口都需要使用含有Authorization Token和Application ID的header, 其他的header信息根据不同的接口在下面进行阐述。

这些Header信息可以帮助我们的系统验证发送请求者的身份信息：

- 验证客户Token，确保发起请求者是我们认证过的外部开发者
- 验证该授权可以访问 v1/resource/{id} 的相关资源
- 将您提供的 x-store-id(如:123) 匹配成我们系统内的store ID ( 如：AAA )
- 验证并读取我们系统内该店铺 ( AAA ) 的所有信息。

*All resource endpoints except the Authorization and X-Application-Id header, the remaining headers are explicitly stated in the individual endpoint documentation section.*

With these headers, the system will:

- Validates the client token, making sure the call is originating from a trusted source.
- Validates that the Application has the permission to access the v1/resource/{id} resource via the Application's pre-configured scopes.
- Translate your X-Store-Id to our internal store ID (e.g. AAA).
- Validate and retrieve the resource AAA that is associated with your Application via store id 321.

POST/PUT模式看起来与GET模式相似, 但是他们需要在发送请求时包含body信息 ( 默认为 application/json内容格式), 如下面的代码示例

POST/PUT methods will look similar to the GET calls, but they'll take in a body in the HTTP request (default to the application/json content-type).

```
curl --location --request POST 'https://partners.cloudkitchens.com/v1/resource' \
--header 'Authorization: Bearer 34fdabeeafds=' --header 'X-Store-Id: 321' --header
'X-Application-Id: e22f94b3-967c-4e26-bf39-9e364066b68b"
--data '{"foo": "bar"}'
```

## 接口授权 API Auth - OAuth2.0

安全模式 Security Scheme Type	OAuth2
客户端密钥授权流程 clientCredentials OAuth Flow	Token URL: /v1/auth/token Scopes: <ul style="list-style-type: none"><li>• <b>orders.create</b> - 发送新订单 Post new orders</li><li>• <b>orders.status_update</b> - 更新订单状态 Update order status</li><li>• <b>menus.read</b> - 读取菜单 Read menus</li><li>• <b>menus.upsert</b> - 上传菜单 Upsert menus</li><li>• <b>ping</b> - 服务器测试 Ping the system</li></ul>

授权API基于OAuth2.0协议，提供客户端密钥发放功能。请求发起者需要在请求的“Authorization” header中包含提供一个有效的token(Bearer)。使用在开通时给您提供的Application ID和Client Secret, 通过token授权接口申请token。申请的结果会是一个含有效期的token (同时该token也有可能被提前手动终止有效)。该接口返回的header中将包含一个Bearer token，同时返回该token的有效期。

注意:下面提到的client\_id是OAuth2.0的协议名称，在这里指的是您开通时的application ID

The Authorization API is based on the OAuth2.0 protocol, using the client credentials grant. Resources expect a valid token sent as a Bearer token in the HTTP Authorization header.

To generate the token, use the Application ID and Client Secret (provided during onboarding) to the Token Auth endpoint. The result of this invocation is a token that is valid for a predetermined time or until it is manually revoked.

The response of the following endpoints will return a token that will be sent as a Bearer value of the Authorization HTTP header, along with meta information such as expiry-date.

*Note that the referred client\_id is the Application ID because though we chose to adhere to the OAuth2.0 standard for the auth APIs.*

## 请求示例 Request Examples

**Post**请求的数据需使用**URL Encoded**格式进行编码

我们的API给您提供一个使用client\_id和client\_secret token创建的接口，使用 *application/x-www-form-urlencoded* 内容格式。

The API exposes a token generation endpoint expecting your *client\_id* and *client\_secret* to be formatted as *application/x-www-form-urlencoded* content type.

```
curl --location --request POST 'https://partners-staging.baitime.cn/v1/auth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'scope=ping' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=[APPLICATION_ID]' \
--data-urlencode 'client_secret=[CLIENT_SECRET]'
```

### HTTP Basic 授权 HTTP Basic Auth

同时，我们也提供另外一种Basic授权header模式，该格式使用Base64编码将client\_id (Application ID)和client\_secret计算出授权所需的内容。

Alternatively, the API also accepts a Basic Authorization header with the Base64 encoding of the client\_id (Application ID) and client\_secret joined by a single colon .:

```
BASE64_ENCODED_CREDENTIALS = base64_encode(client_id + ":" + client_secret)
```

```
curl --location --request POST 'https://partners-staging.baitime.cn/v1/auth/token' \
--header 'Authorization: Basic [BASE64_ENCODED_CREDENTIALS]' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'scope=ping' \
--data-urlencode 'grant_type=client_credentials'
```

# 回调模式 Webhook

我们的开放平台可以通过HTTP POST方式给您的系统发送消息，同时包含了 HMAC digest签名（在回调授权部分可以找到相应内容）。请联系我们的团队帮助您创建回调模式。

The API platform is able to send notifications to your system via HTTP POST requests signed with an HMAC digest defined in the [Webhook AUTH section](#). Please work with your Account Representative to setup your Application's Webhook configurations.

```
Example Base-URL = https://yourserver.com/webhook
```

目前，我们支持以下几种事件消息

Currently, supports the following notification events:

Notification event 事件消息	Description 描述	Metadata Schema metadata模式
orders.new_order	当订单创建时发送消息 Sent when an order is created.	<a href="#">Order</a>
orders.order_status_update	当订单状态更新时发送消息 Sent when order status is updated	<a href="#">Order Status History</a>

## 消息格式 Notification Schema

Name 名称	Type 类型	Description 描述
eventId	string	独特的事件ID Unique id of the event.
eventTime	string	事件发生时间 The time the event occurred.
eventType	string	事件类型 The type of event (e.g. create_order).
metadata.storeId	string	事件发生的店铺ID Id of the store for which the event is being published.



metadata.applicationId	string	事件相关通知方的application ID Id of the application for which the event is being published.
metadata.resourceId	string	该事件发送到的外部数据源的标识 The external identifier of the resource that this event refers to.
metadata.resourceHref	string	调用数据源的详细信息接口 The endpoint to fetch the details of the resource.
metadata.payload	object	各种事件的对象 (在回调模式里有详细描述) The event object which will be detailed in each Webhook description.

## 消息请求示例 **Notification Request Example**

```
curl --location --request POST 'https://yourserver.com/webhook' \
--header 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36' \
--header 'Authorization: MAC <hash signature>' \
--header 'Content-Type: application/json' \
--data-raw '{
  "eventId": "123456",
  "eventTime": "2020-10-10T20:06:02:123Z",
  "eventType": "orders.new_order",
  "metadata": {
    "storeId": "755fd19a-7562-487a-b615-171a9f89d669",
    "applicationId": "e22f94b3-967c-4e26-bf39-9e364066b68b",
    "resourceHref":
"https://partners-staging.api.com/v1/orders/bf9fd81-f213-496e-a026-91b6af44996c",
    "resourceId": "bf9fd81-f213-496e-a026-91b6af44996c",
    "payload": {}
  }
}
```

## 回调授权 **Webhook Auth**

回调模式会需要您的系统使用HMAC签名来验证该请求是从我们系统发出。

包含该签名的授权header用来验证信息是我们发送，用来生成digest的client\_secret (开通时提供的)是您与我们共享的

Webhooks use a standard HMAC signature for your system to validate that the inbound request is coming from our API.

The sent in the Authorization header should be used to validate that the payload was sent by us. The shared `client_secret` (configured during onboarding) is used to generate the digest:

Python:

```
// payload_body = Body of the payload.
// client_secret = The shared secret key.

hash_signature = getHashSignatureFromAuthHeader() // hash_signature will be
prefixed with "MAC "

hash_signature_digest = hash_signature.split(" ")[1]; // strip off the prefix "MAC "

hash_signature_digest_decoded = Base64.decode(hash_signature_digest, UTF_8)

digerster = hmac.new(client_secret, payload_body, "HmacSHA1");

assert hash_signature_digest_decoded == digester.hexDigest()
```

Java:

```
// payloadBody = Body of the payload.
// clientSecret = The shared secret key.

String hashSignature = getHashSignatureFromAuthHeader(); // hash_signature will be
prefixed with "MAC "

String hashSignatureDigest = hashSignature.split(" ")[1]; // strip off prefix "MAC "

byte[] hashSignatureDigestDecoded =
Base64.getDecoder().decode(hashSignatureDigest);

Mac mac = Mac.getInstance(HMAC_SHA_1);

mac.init(new SecretKeySpec(clientSecret.getBytes(Charsets.UTF_8), HMAC_SHA_1));
```

```
assertArrayEquals(hashSignatureDigestDecoded,  
mac.doFinal(payloadBody.getBytes()));
```

# 接口 Endpoints

## 授权 Auth

授权管理相关的接口 Endpoints to handle token management.

## 生成token Generate token

在request body中和HTTP basic Auth中提供客户端密钥都是可以支持的。

Client credentials in the request-body and HTTP Basic Auth are supported.

**Request body模式: REQUEST BODY SCHEMA: application/x-www-form-urlencoded**

client_id	string <uuid> 客户端的application id ( 账号开通时提供的 ) The ID of the client (also known as the Application ID).
client_secret	string 客户端的secret ( 账号开通时提供的 ) The secret of the client.
grant_type Required (必填)	string Value: "client_credentials" 授权的种类 - 支持OAuth2.0发放模式 The OAuth2.0 grant types are supported.
scope Required (必填)	string 授权的内容范围。多种范围用空格来进行区分 The scope to request multiple scopes are passed delimited by a space character.  Scopes: <ul style="list-style-type: none"><li>• <b>orders.create</b> - 发送新订单 Post new orders</li></ul>

---

- `orders.status_update` - 更新订单状态 Update order status
  - `menus.read` - 读取菜单 Read menus
  - `menus.upsert` - 上传菜单 Upsert menus
  - `ping` - 服务器测试 Ping the system
- 

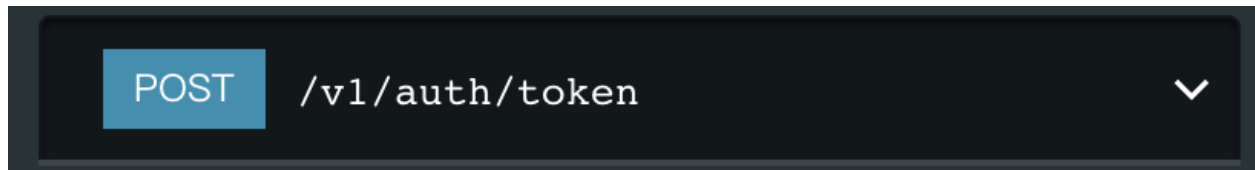
## 返回信息 Responses

**200 Token成功生成** **The token was successfully generated.**

**400 请求格式错误** **The request is malformed.**

**401 无效的验证信息** **Invalid authorization.**

**422 请求的body内容无效** **The request body is not valid.**



<https://partners-staging.baitime.cn/v1/auth/token>

Content type: application/json

Status: 200

```
{
  "access_token": "string",
  "expires_in": 0,
  "scope": "string",
  "token_type": "string"
}
```

Status: 400/401/422

```
{
  "message": "The request body is invalid."
}
```

# 菜单 Menus

菜单管理类的接口 Endpoints to manage menus.

## 读取某个门店的菜单 **Get the menus for a store**

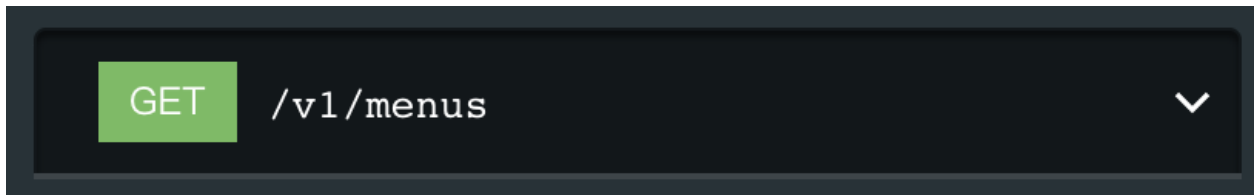
授权 AUTHORIZATIONS: [OAuth2.0](#) (menus.read)

Header参数: HEADER PARAMETERS

X-Application-Id Required (必填)	string Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 客户端的application ID ( 账号开通时提供的 ) The application ID of the client.
X-Store-Id Required (必填)	string 举例 : 1 一个一家门店的唯一标识。这个信息必须在所有对接信息中是唯一的。 Example: 1 A unique identifier of the store. This value must be unique within the application.

## 返回 Responses

- 200** 门店的菜单 **The store's menu.**
- 400** 请求格式错误 **The request is malformed.**
- 401** 无效的验证信息 **Invalid authorization.**
- 403** 未授权 **Authorization not valid for requested entity.**
- 404** 请求的店铺无法找到 **Requested entity not found.**
- 422** 请求的body内容无效 **The request body is not valid.**



<https://partners-staging.baitime.cn./v1/menus>

Content type: application/json

**Status: 200**

```
{
  "menus": {
    "1c753dac-5b48-11eb-ae93-0242ac130002": {
      "name": "Tasty BBQ",
      "description": "Cooking up BBQ deliciousness from around the globe!",
      "id": "ff6dd693-5e55-4a92-a359-ea61b23ed423",
      "categoryIds": [
        "string"
      ],
      "hours": {
        "intervals": [
          {
            "day": "Monday",
            "fromHour": 8,
            "fromMinute": 0,
            "toHour": 20,
            "toMinute": 45
          }
        ]
      }
    }
  },
  "items": {
    "2c75405e-5b48-11eb-ae93-0242ac130002": {
      "id": "faa4c79f-480d-4de1-bc34-5fb74ef082ef",
      "name": "Bagel",
      "description": "Delicious bagel!",
      "price": {
        "currencyCode": "EUR",
        "amount": 25.21
      },
      "priceOverrides": "$3.50 for UberEats, $4 for DoorDash.",
      "status": {
        "saleStatus": "TEMPORARILY_NOT_FOR_SALE",
        "suspendedUntil": "2020-11-23T21:33:51Z"
      },
      "modifierGroupIds": [
        "string"
      ]
    }
  }
}
```

```
    ],
    "photoIds": [
      "string"
    ],
    "photoUrls": [
      "string"
    ]
  }
},
"photos": {
  "3c754180-5b48-11eb-ae93-0242ac130002": {
    "id": "e9174f75-a293-4908-bba7-9db69871ad81",
    "contentType": "PNG",
    "url":
"https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_272x92dp.png"
  }
},
"categories": {
  "4c7542c0-5b48-11eb-ae93-0242ac130002": {
    "name": "Breakfast",
    "description": "All early morning items served up hot and fresh!",
    "id": "da0e4e94-5670-4175-897a-3b7dde45bed5",
    "itemIds": [
      "string"
    ]
  }
},
"modifierGroups": {
  "5c7543b0-5b48-11eb-ae93-0242ac130002": {
    "id": "db5740de-429c-4f06-b264-3d41dce6d8c3",
    "name": "Choose your type of bagel",
    "description": "Choose any of these delicious types of bagels",
    "minimumSelections": 1,
    "maximumSelections": 1,
    "itemIds": [
      "string"
    ]
  }
}
}
```

```
}
```

Status: 400/401/403/404/422

```
{  
  "message": "The request body is invalid."  
}
```

## 菜单上传 Upsert menus for a store

AUTHORIZATIONS: OAuth2.0 (menus.upsert)

Header参数: HEADER PARAMETERS

X-Application-Id required string  
Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4  
客户端的application ID ( 开通时提供的 )  
The application ID of the client.

X-Store-Id required string  
Example: 1  
店铺的唯一标识。这个信息必须在所有对接信息中是唯一的。  
A unique identifier of the store. This value must be unique within the application.

## Request Body的模式: REQUEST BODY SCHEMA: application/json

字段	描述	字段	描述	字段	描述	字段	描述
menus	object (所有菜单, 以菜单ID为序号) 门店菜单	name	string 菜单名称				
		description	string 菜单描述				
		id	string <span style="color: red;">required</span> 菜单ID				
		categoryIds	Array of strings <span style="color: red;">required</span> 品类ID列表				



hours	object (Hours) 菜单生效时期	intervals required	Array of objects (HourInterval) 菜单起止时间列表	day required  fromHour required  fromMinute required  toHour required  toMinute required	string Enum: "MONDAY" "TUESDAY" "WEDNESDAY" "THURSDAY" "FRIDAY" "SATURDAY" "SUNDAY" 星期几的标识  integer <int32> 菜单生效小时  integer <int32> 菜单生效分钟  integer <int32> 菜单结束小时  integer <int32> 菜单结束分钟	
items required	object (All Items for the store, indexed by ID) 菜单上售卖的菜品。菜品的规格也是单独的菜品，他们与绑定的菜品关系被定义为规格组	id required	string 该菜品的ID			
		name required	string 菜品名称			
		description required	string 菜品描述			
		price required	object (ItemPrice) 菜品的价格，必须提供币种信息。	currencyCode required  amount	string 3 characters 三个字符的币种信息：人民币用CNY  number 菜品价格	
		priceOverrides	Array of objects (ItemPriceOverride) 特殊平台的价格覆盖	rules required  currencyCode required  amount	Array of objects (OverrideRule) 价格覆盖的规则  string 3 characters 三个字符的币种信息：人民币用CNY  number 菜品价格	externalServiceSlug required  string 该覆盖规则对哪个平台生效
status required	object (ItemStatus) 该菜品售卖的状态 (如：在售，暂下架，永远下架)	saleStatus required  suspendedUntil	string Enum: "FOR_SALE" "INDEFINITELY_NOT_FOR_SALE" "TEMPORARILY_NOT_FOR_SALE" 售卖状态  string <date-time> 标志该菜品停售到期的时间戳			
modifierGroupIds required	Array of strings 所属该菜品的各规格组的ID列表					
photoIds required	Array of strings 菜品相关的照片ID列表					

		photoUrIs	Array of strings 菜品照片URL列表。该字段只用于POST/PUT的请求		
photos required	object (All ItemPhotos for the store, indexed by ID) 显示菜品照片需要的所有信息	id	string 菜品照片的ID		
		contentType	string 照片内容格式		
		url required	string 照片URL		
categories required	object (All Categories for the store, indexed by ID) 菜品品类	name required	string 品类名称		
		description	string 品类描述		
		id required	string 品类的ID		
		itemIds required	Array of strings 该品类所有的菜品ID		
modifierGroups required	object (All ModifierGroups for the Store, indexed by ID) 菜品规格组。规格也是一个菜品, 与主菜品关系为所属关系	id required	string 规格组的ID		
		name required	string 规格组的名称 (如:杯型)		
		description	string 规格组的描述		
		minimumSelections	integer <int32> 该规格组最小选择数		
		maximumSelections	integer <int32> 该规格组最大选择数		
		itemIds	Array of strings 该规格组包含的各个菜品的ID列表		

## 返回 Responses

**200** 门店的菜单 **The store's menu.**

**400** 请求格式错误 **The request is malformed.**

**401** 无效的验证信息 **Invalid authorization.**

**403** 未授权 **Authorization not valid for requested entity.**

**404** 请求的店铺无法找到 **Requested entity not found.**

**422** 请求的body内容无效 **The request body is not valid.**

示例

POST /v1/menus

<https://partners-staging.baitime.cn/v1/menus>

Content type: application/json

Request Payload Samples

```
{
  "menus": {
    "1c753dac-5b48-11eb-ae93-0242ac130002": {
      "name": "Tasty BBQ",
      "description": "Cooking up BBQ deliciousness from around the globe!",
      "id": "ff6dd693-5e55-4a92-a359-ea61b23ed423",
      "categoryIds": [
        "string"
      ],
      "hours": {
        "intervals": [
          {
            "day": "Monday",
            "fromHour": 8,
            "fromMinute": 0,
            "toHour": 20,
            "toMinute": 45
          }
        ]
      }
    }
  },
  "items": {
    "2c75405e-5b48-11eb-ae93-0242ac130002": {
      "id": "faa4c79f-480d-4de1-bc34-5fb74ef082ef",
      "name": "Bagel",
      "description": "Delicious bagel!",
      "price": {
        "currencyCode": "EUR",
        "amount": 25.21
      }
    }
  }
}
```

```
    },
    "priceOverrides": "$3.50 for UberEats, $4 for DoorDash.",
    "status": {
      "saleStatus": "TEMPORARILY_NOT_FOR_SALE",
      "suspendedUntil": "2020-11-23T21:33:51Z"
    },
    "modifierGroupIds": [
      "string"
    ],
    "photoIds": [
      "string"
    ],
    "photoUrls": [
      "string"
    ]
  }
},
"photos": {
  "3c754180-5b48-11eb-ae93-0242ac130002": {
    "id": "e9174f75-a293-4908-bba7-9db69871ad81",
    "contentType": "PNG",
    "url":
"https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_272x92dp.png"
  }
},
"categories": {
  "4c7542c0-5b48-11eb-ae93-0242ac130002": {
    "name": "Breakfast",
    "description": "All early morning items served up hot and fresh!",
    "id": "da0e4e94-5670-4175-897a-3b7dde45bed5",
    "itemIds": [
      "string"
    ]
  }
},
"modifierGroups": {
  "5c7543b0-5b48-11eb-ae93-0242ac130002": {
    "id": "db5740de-429c-4f06-b264-3d41dce6d8c3",
    "name": "Choose your type of bagel",
    "description": "Choose any of these delicious types of bagels",
    "minimumSelections": 1,
    "maximumSelections": 1,
    "itemIds": [
      "string"
    ]
  }
}
}
```

**Status: 202**

```
{
  "menus": {
    "1c753dac-5b48-11eb-ae93-0242ac130002": {
      "name": "Tasty BBQ",
      "description": "Cooking up BBQ deliciousness from around the globe!",
      "id": "ff6dd693-5e55-4a92-a359-ea61b23ed423",
      "categoryIds": [
        "string"
      ],
      "hours": {
        "intervals": [
          {
            "day": "Monday",
            "fromHour": 8,
            "fromMinute": 0,
            "toHour": 20,
            "toMinute": 45
          }
        ]
      }
    }
  },
  "items": {
    "2c75405e-5b48-11eb-ae93-0242ac130002": {
      "id": "faa4c79f-480d-4de1-bc34-5fb74ef082ef",
      "name": "Bagel",
      "description": "Delicious bagel!",
      "price": {
        "currencyCode": "EUR",
        "amount": 25.21
      },
      "priceOverrides": "$3.50 for UberEats, $4 for DoorDash.",
      "status": {
        "saleStatus": "TEMPORARILY_NOT_FOR_SALE",
        "suspendedUntil": "2020-11-23T21:33:51Z"
      },
      "modifierGroupIds": [
        "string"
      ],
      "photoIds": [
        "string"
      ],
    }
  }
}
```

```
"photoUrls": [
  "string"
]
},
"photos": {
  "3c754180-5b48-11eb-ae93-0242ac130002": {
    "id": "e9174f75-a293-4908-bba7-9db69871ad81",
    "contentType": "PNG",
    "url":
"https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_272x92dp.png"
  }
},
"categories": {
  "4c7542c0-5b48-11eb-ae93-0242ac130002": {
    "name": "Breakfast",
    "description": "All early morning items served up hot and fresh!",
    "id": "da0e4e94-5670-4175-897a-3b7dde45bed5",
    "itemIds": [
      "string"
    ]
  }
},
"modifierGroups": {
  "5c7543b0-5b48-11eb-ae93-0242ac130002": {
    "id": "db5740de-429c-4f06-b264-3d41dce6d8c3",
    "name": "Choose your type of bagel",
    "description": "Choose any of these delicious types of bagels",
    "minimumSelections": 1,
    "maximumSelections": 1,
    "itemIds": [
      "string"
    ]
  }
}
}
```

**Status: 400/401/403/404/422**

```
{
  "message": "The request body is invalid."
}
```

获取菜单同步任务状态 **Get the async menu job status.**

AUTHORIZATIONS: OAuth2.0 (menus.async\_job.read)

Path参数: PATH PARAMETERS

jobId	string
Required (必填)	Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 唯一的任务ID The unique identifier of the job.

---

Header参数: HEADER PARAMETERS

X-Application-Id	string
Required (必填)	Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 客户端的application ID ( 账号开通时提供的 ) The application ID of the client.

---

X-Store-Id	string
Required (必填)	Example: 1 门店的唯一标识。这个信息必须在所有对接信息中是唯一的。 A unique identifier of the store. This value must be unique within the application.

---

返回 Responses

- 200** 该菜单同步任务的状态 **The menu async job status.**
- 400** 请求格式错误 **The request is malformed.**
- 401** 无效的验证信息 **Invalid authorization.**
- 403** 未授权 **Authorization not valid for requested entity.**
- 404** 请求的店铺无法找到 **Requested entity not found.**
- 422** 请求的body内容无效 **The request body is not valid.**

示例

```
GET /v1/menus/jobs/{jobId} ▾
```

<https://partners-staging.baitime.cn/v1/menus/jobs/{jobId}>

### Status: 200

```
{
  "jobReference": {
    "id": "497f6eca-6276-4993-bfeb-53cbbbba6f08",
    "status": "PENDING"
  }
}
```

### Status: 400/401/403/404/422

```
{
  "message": "The request body is invalid."
}
```

## 订单接口 **Orders**

用来管理门店订单的接口 Endpoints to manage orders for a store.

### 订单创建 **Create order**

AUTHORIZATIONS: [OAuth2.0](#) (orders.create)

### Header参数: **HEADER PARAMETERS**

X-Application-Id  
**required**

string  
Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4  
客户端的application ID ( 账号开通时提供的 )  
The application ID of the client.

---

X-Store-Id  
**required**

string  
Example: 1  
门店的唯一标识。这个信息必须在所有对接信息中是唯一的。  
A unique identifier of the store. This value must be unique within the application.

---

请求Body模式: **REQUEST BODY SCHEMA**application/json



字段名称	描述	字段名称	描述	字段名称	描述	字段名称	描述
externalIdentifiers	object 订单ID <b>required</b>	id	string 唯一的订单ID				
		friendlyId	string 骑手取餐使用的订单ID				
items	Array of objects (Item) Items ordered.	id	string 菜品唯一ID				
		name	string 顾客看到的菜品名称				
		quantity	integer <int32> >= 1 <b>required</b> 菜品份数				
		note	string 顾客对菜品的备注				
		categoryId	string 菜品品类ID				
		categoryName	string 菜品品类名称				
		price	number 菜品价格				
		modifiers	Array of objects (ItemModifier) 菜品规格	id	string 规格ID		
				name	string 规格名称 (如: 大杯)		
				quantity	integer <int32> >= 1 <b>required</b> 规格数量		
				price	number 规格价格		
				groupName	string 规格组名称 (如: 杯型)		
				groupId	string 规格组ID		
				modifiers	Array of objects (ItemModifier) 如有多级规格, 请循环以上		
orderedAt	string <date-time> 顾客下单时间 (UTC时区)						

currencyCode <b>required</b>	string 3 characters 3位币种。人民币用CNY				
customer	object (Person) 顾客信息	name <b>required</b>	string 顾客姓名		
		phone	string 顾客电话		
		email	string 顾客email		
		personalIdentifiers	object 顾客个人标识	taxIdentificationNumber	string 用户Tax ID
customerNote	string 顾客订单备注				
status <b>required</b>	string Enum: "NEW_ORDER" "CONFIRMED" "PICKED_UP" "CANCELED" "FULFILLED" "PREPARED" "UNKNOWN" 订单状态				
deliveryInfo	object 订单配送信息	courier	object 骑手信息	name <b>required</b>	string 骑手姓名
				phone	string 骑手电话
				email	string 骑手email
				personalIdentifiers	object 骑手的ID
				taxIdentificationNumber	string 骑手的Tax ID
		destination	object (Address) 顾客地址	postalCode	string 地址邮编
				city	string 城市名称
				state	string 州/省名称
				countryCode	string 国家编码 <a href="http://cldr.unicode.org/">http://cldr.unicode.org/</a>
				addressLines	Array of strings 地址详情
				location	object 顾客下单位置
				latitude <b>required</b>	number <double> 地址的经度
				longitude <b>required</b>	number <double> 地址的纬度
		licensePlate	string 骑手车牌号		
		makeModel	string 骑手车型号		

		lastKnownLocation	object (Location) 最新骑手位置	latitude <b>required</b>	number <double> 位置的经度	
				longitude <b>required</b>	number <double> 位置的纬度	
		note	string 配送备注			
orderTotal <b>required</b>	object (OrderTotal) 订单金额信息	subtotal <b>required</b>	number 订单所有菜品原价总和			
		discount	number 订单折扣			
		tax	number 顾客付的订单税额			
		tip	number 骑手小费			
		deliveryFee	number 配送费			
		total	number 订单最终实付金额			
		couponCode	string 优惠券使用信息			
customerPayments	Array of objects (CustomerPayment) 顾客付款信息	value <b>required</b>	number 应付金额			
		processingStatus <b>required</b>	string Enum: "COLLECTABLE" "PROCESSED" 付款状态			
		paymentMethod <b>required</b>	string Enum: "CASH" "CARD" 付款方式			
fulfillmentInfo	object (FulfillmentInfo) 订单交付信息	pickupTime	string <date-time> 预计取餐时间 (UTC)			
		deliveryTime	string <date-time> 预计送达时间			
		fulfillmentMode	string Enum: "DELIVERY" "RESTAURANT_DELIVERY" "PICKUP" 订单交付方式			

	schedulingType	string Enum: "ASAP" "FIXED_TIME" 预订单种类。如果该 字段为空我们默认订 单是实时订单		
	courierStatus	string Enum: "COURIER_ASSIG NED" "COURIER_ON_R OUTE_TO_PICKUP " "COURIER_ARRIV ED" "COURIER_PICKE D_UP_FOOD" "COURIER_COMP LETED" 骑手状态		

## 返回 Responses

**200** 订单成功创建 **The order was successfully created.**

**400** 请求格式错误 **The request is malformed.**

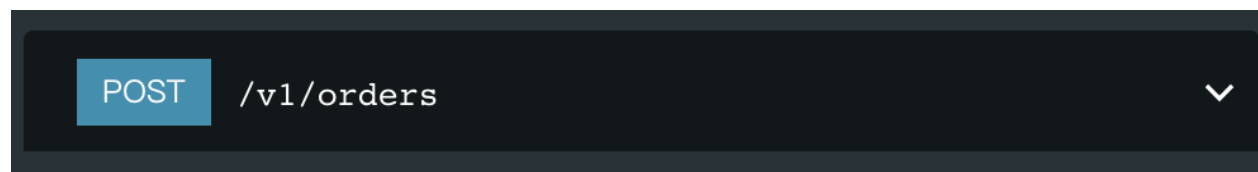
**401** 无效的验证信息 **Invalid authorization.**

**403** 未授权 **Authorization not valid for requested entity.**

**404** 请求的店铺无法找到 **Requested entity not found.**

**422** 请求的body内容无效 **The request body is not valid.**

## 示例



<https://partners-staging.baitime.cn/v1/orders>

Content type: application/json

Request Payload sample

```
{
  "externalIdentifiers": {
    "id": "69f60a06-c335-46d9-b5a1-97f1a211c514",
  }
}
```

```
"friendlyId": "ABCDE"
},
"items": [
  {
    "id": "33e0418f-3d56-4360-ba03-18fc5f8844a3",
    "name": "Juicy Cheeseburger",
    "quantity": 1,
    "note": "Please cook to well done!",
    "categoryId": "303de078-870d-4349-928b-946869d4d69b",
    "categoryName": "Burgers",
    "price": 5.9,
    "modifiers": [
      {
        "id": "d7a21692-9195-43aa-a58f-5395bba8a804",
        "name": "Avocado",
        "quantity": 1,
        "price": 1,
        "groupName": "Add ons",
        "groupId": "fb52b138-7ac4-42c1-bfd8-664d57113a41",
        "modifiers": [
          {}
        ]
      }
    ]
  }
],
"orderedAt": "2020-12-20T02:30:00Z",
"currencyCode": "EUR",
"customer": {
  "name": "Jane Doe",
  "phone": "+1-555-555-5555",
  "email": "email@email.com",
  "personalIdentifiers": {
    "taxIdentificationNumber": "01234567890"
  }
},
"customerNote": "Please include extra napkins!",
"status": "NEW_ORDER",
"deliveryInfo": {
  "courier": {
    "name": "Jane Doe",
    "phone": "+1-555-555-5555",
    "email": "email@email.com",
    "personalIdentifiers": {
      "taxIdentificationNumber": "01234567890"
    }
  }
},
"destination": {
  "postalCode": "20500",
```

```
"city": "Washington",
"state": "DC",
"countryCode": "US",
"addressLines": "1600 Pennsylvania Avenue NW",
"location": {
  "latitude": 38.8977,
  "longitude": 77.0365
}
},
"licensePlate": "ABC 123",
"makeModel": "Honda CR-V",
"lastKnownLocation": {
  "latitude": 38.8977,
  "longitude": 77.0365
},
"note": "Gate code 123"
},
"orderTotal": {
  "subtotal": 11.97,
  "discount": 1,
  "tax": 1.1,
  "tip": 2,
  "deliveryFee": 5,
  "total": 19.07,
  "couponCode": "VWXYZ98765"
},
"customerPayments": [
  {
    "value": 2,
    "processingStatus": "COLLECTABLE",
    "paymentMethod": "CASH"
  }
],
"fulfillmentInfo": {
  "pickupTime": "2020-12-20T02:30:00Z",
  "deliveryTime": "2020-12-20T02:30:00Z",
  "fulfillmentMode": "DELIVERY",
  "schedulingType": "ASAP",
  "courierStatus": "COURIER_ASSIGNED"
}
}
```

### Status:200

```
{
  "externalIdentifiers": {
    "source": "ubereats",
    "id": "69f60a06-c335-46d9-b5a1-97f1a211c514",
    "friendlyId": "ABCDE"
  }
}
```

```
},  
"storeId": "string"  
}
```

### Status:400/401/403/404/422

```
{  
  "message": "The request body is invalid."  
}
```

## 订单状态更新 **Update order status**

AUTHORIZATIONS: [OAuth2.0](#) (orders.status\_update)

Path参数: PATH PARAMETERS

orderId	string
Required (必填)	Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 该订单唯一的标识，uuid的格式 A unique identifier of the order in a UUID format.

---

### Header参数 **HEADER PARAMETERS**

X-Application-Id	string
Required (必填)	Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 客户端的application ID ( 账号开通时提供的 ) The application ID of the client.

---

X-Store-Id	string
Required (必填)	Example: 1 门店的唯一标识。这个信息必须在所有对接信息中是唯一的。 A unique identifier of the store. This value must be unique within the application.

---

Request Body模式 **REQUEST BODY SCHEMA: application/json**

orderStatus string  
Enum: "PREPARED" "CANCELED" "FULFILLED"  
订单变更后的状态。包含：备餐完成，已取消，已送达  
The requested status to transition the order to.

---

## 返回 Responses

**202 订单状态成功更新 The order status was successfully accepted.**

**400 请求格式错误 The request is malformed.**

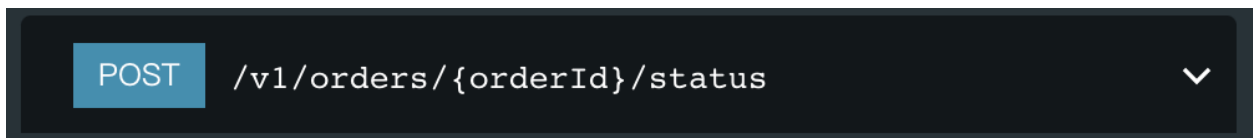
**401 无效的验证信息 Invalid authorization.**

**403 未授权 Authorization not valid for requested entity.**

**404 请求的店铺无法找到 Requested entity not found.**

**422 请求的body内容无效 The request body is not valid.**

示例



<https://partners-staging.baitime.cn/v1/orders/{orderId}/status>

Content type: application/json

Request Payload Sample:

```
{  
  "orderStatus": "PREPARED"  
}
```

Status: 202

```
{}
```

Status: 400/401/403/404/422

```
{  
  "message": "The request body is invalid."  
}
```



# 服务器测试 Ping

用来进行测试服务器连接和系统验证的接口 Endpoints to ping and test system authentication.

## 发送测试 Ping the system

AUTHORIZATIONS: OAuth2.0 (ping)

### Header参数: HEADER PARAMETERS

X-Application-Id <b>Required</b> (必填)	string Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 客户端的application ID ( 账号开通时提供的 ) The application ID of the client.
X-Store-Id <b>Required</b> (必填)	string Example: 1 门店的唯一标识。这个信息必须在所有对接信息中是唯一的。 A unique identifier of the store. This value must be unique within the application.
X-Echo-Error	string Example: ping test error message 接口返回的错误信息，用来测试使用。 The error message to be returned by the endpoint, for testing purposes.

## 返回 Responses

**200 服务器测试成功 The ping was successfully received.**

**401 授权信息无效 Invalid authorization.**

**418 I am a teapot.**

示例



<https://partners-staging.baitime.cn/v1/ping>

Content type: application/json

Status: 200

```
{
  "response": "string",
  "currentTime": "2019-08-24T14:15:22Z"
}
```

Status: 401/418

```
{
  "message": "The request body is invalid."
}
```

## 配送接口 **Delivery**

用来管理订单配送的接口 Endpoints to manage delivery.

### 请求配送 **Request delivery quote event result callback endpoint**

AUTHORIZATIONS: [OAuth2.0](#) (delivery.provider)

#### Header参数: **HEADER PARAMETERS**

X-Application-Id Required (必填)	string Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 The plain-text Application Id, provided at onboarding.
--------------------------------------	---

---

X-Store-Id Required (必填)	string Example: 1 The ID of the store in your system this call acts on behalf of.
--------------------------------	---

---

X-Event-Id Required (必填)	string Example: cf0ce51b-d74e-40d3-b177-1925ab4edc0c Unique identifier of the event that this callback refers to.
-----------------------------	---

---

## Request Body 模式 REQUEST BODY SCHEMA: application/json

minPickupDuration      integer <int32> >= 0  
Minimum time required for courier to arrive at pickup location in minutes It is an estimation.

---

maxPickupDuration      integer or null <int32>  
Maximum time that the courier's arrival at the pick up location can be delayed. If not provided, we will treat it as if there is no limit. This value is an estimation and expressed in minutes.

---

deliveryDistance      object or null (Distance)  
Delivery distance.

---

currencyCode      string 3 characters  
The 3-letter currency code (ISO 4217) to use for all monetary values.

---

cost      object (DeliveryCost)  
Delivery cost details.

---

provider      string or null  
Delivery Service Provider Slug.

---

fulfillmentPath      Array of objects or null (FulfillmentPathEntity)  
List of entities involved in the fulfillment processing path.

---

createdAt      string <date-time>  
The time that the quote was created.

---

## 返回 Responses

**204** 事件回调被成功接受 **The event callback was successfully accepted.**

**400** 请求格式错误 **The request is malformed.**

**401** 无效的验证信息 **Invalid authorization.**

**403** 未授权 **Authorization not valid for requested resource.**

**404** 请求的信息无法找到 **Resource not found.**

**422** 请求的body内容无效 **The request body is not valid.**

示例

```
POST /v1/delivery/{deliveryReferenceId... ▾
```

<https://partners-staging.baitime.cn/v1/delivery/{deliveryReferenceId}/quotes>

Content type: application/json

Request Payload Sample:

```
{
  "minPickupDuration": 5,
  "maxPickupDuration": 10,
  "deliveryDistance": {
    "unit": "KILOMETERS",
    "value": 0
  },
  "currencyCode": "EUR",
  "cost": {
    "baseCost": 4.99,
    "extraCost": 0.99
  },
  "provider": "string",
  "fulfillmentPath": [
    {}
  ],
  "createdAt": "2007-12-03T10:15:30+01:00"
}
```

Status: 400/401

```
{
  "message": "The request body is invalid."
  "details": "The error validation details."
}
```

**接受配送 Accept delivery event result endpoint**

AUTHORIZATIONS: OAuth2.0 (delivery.provider)

## Header参数: HEADER PARAMETERS

X-Application-Id Required (必填)	string Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 The plain-text Application Id, provided at onboarding.
X-Store-Id Required (必填)	string Example: 1 The ID of the store in your system this call acts on behalf of.
X-Event-Id Required (必填)	string Example: cf0ce51b-d74e-40d3-b177-1925ab4edc0c Unique identifier of the event that this callback refers to.

## Request Body模式 REQUEST BODY SCHEMA: application/json

deliveryDistance	object or null (Distance) Delivery distance.
currencyCode	string 3 characters The 3-letter currency code (ISO 4217) to use for all monetary values.
cost	object (DeliveryCost) Delivery cost details.
fulfillmentPath	Array of objects or null (FulfillmentPathEntity) List of entities involved in the fulfillment processing path.
estimatedDeliveryTime	string or null <date-time> The time that the quote was created.

---

estimatedPickupTime	string or null <date-time> The time that the quote was created.
confirmedAt	string <date-time> The time that the quote was created.

---

## 返回 Responses

**204 事件回调被成功接受 The event callback was successfully accepted.**

**400 请求格式错误 The request is malformed.**

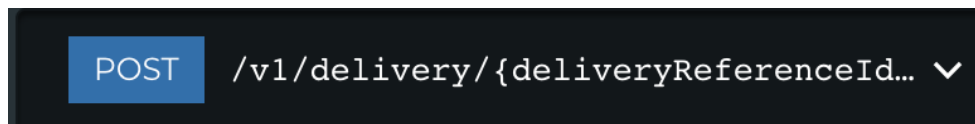
**401 无效的验证信息 Invalid authorization.**

**403 未授权 Authorization not valid for requested resource.**

**404 请求的信息无法找到 Resource not found.**

**422 请求的body内容无效 The request body is not valid.**

示例



<https://partners-staging.baitime.cn/v1/delivery/{deliveryReferenceId}/accept>

Content type: application/json

Request Payload Sample:

```
{
  "deliveryDistance": {
    "unit": "KILOMETERS",
    "value": 0
  },
  "currencyCode": "EUR",
  "cost": {
    "baseCost": 4.99,
    "extraCost": 0.99
  },
  "fulfillmentPath": [
    {
      "name": "string",
      "type": "FULFILLMENT_PROCESSOR"
    }
  ]
}
```

```
}
],
"estimatedDeliveryTime": "2007-12-03T10:15:30+01:00",
"estimatedPickupTime": "2007-12-03T10:15:30+01:00",
"confirmedAt": "2007-12-03T10:15:30+01:00"
}
```

Status: 400/401

```
{
  "message": "The request body is invalid."
  "details": "The error validation details."
}
```

## 取消配送 **Cancel delivery event result endpoint**

AUTHORIZATIONS: [OAuth2.0](#) (delivery.provider)

### Header参数: **HEADER PARAMETERS**

X-Application-Id Required (必填)	string Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4 The plain-text Application Id, provided at onboarding.
--------------------------------------	---

---

X-Store-Id Required (必填)	string Example: 1 The ID of the store in your system this call acts on behalf of.
--------------------------------	---

---

X-Event-Id Required (必填)	string Example: cf0ce51b-d74e-40d3-b177-1925ab4edc0c Unique identifier of the event that this callback refers to.
-----------------------------	---

---

**Request Body**模式 **REQUEST BODY SCHEMA: application/json**

cancelledAt

string <date-time>

The time that the request was cancelled.

---

## 返回 Responses

**204 事件回调被成功接受 The event callback was successfully accepted.**

**400 请求格式错误 The request is malformed.**

**401 无效的验证信息 Invalid authorization.**

**403 未授权 Authorization not valid for requested resource.**

**404 请求的信息无法找到 Resource not found.**

**422 请求的body内容无效 The request body is not valid.**

示例

```
POST /v1/delivery/{deliveryReferenceId... ▾
```

<https://partners-staging.baitime.cn/v1/delivery/{deliveryReferenceId}/cancel>

Content type: application/json

Request Payload Sample:

```
{  
  "cancelledAt": "2007-12-03T10:15:30+01:00"  
}
```

Status: 400/401

```
{  
  "message": "The request body is invalid."  
  "details": "The error validation details."  
}
```

## 更新配送状态 Update delivery status endpoint



AUTHORIZATIONS: [OAuth2.0](#) (delivery.provider)

## Header参数 : HEADER PARAMETERS

X-Application-Id string  
**Required**  
**(必填)** Example: 295f76b4-5725-4bf5-a8ab-97943dbdc3b4  
The plain-text Application Id, provided at onboarding.

---

X-Store-Id string  
**Required**  
**(必填)** Example: 1  
The ID of the store in your system this call acts on behalf of.

---

## Request Body模式 REQUEST BODY SCHEMA: application/json

deliveryStatus string (DeliveryStatus)  
Enum: "REQUESTED" "ALLOCATED" "PICKED\_UP" "COMPLETED"  
"CANCELED"  
The status of the delivery.

---

estimatedDeliveryTime string or null <date-time>  
The time that the quote was created.

---

estimatedPickupTime string or null <date-time>  
The time that the quote was created.

---

courier object or null (Person)  
The recipient information.

---

location object or null (Location)  
Latitude and longitude of the address.

---

createdAt string <date-time>  
The time that the quote was created.

---

## 返回 Responses

**204 事件回调被成功接受 The event callback was successfully accepted.**

**400 请求格式错误 The request is malformed.**

**401 无效的验证信息 Invalid authorization.**

**403 未授权 Authorization not valid for requested resource.**

**404 请求的信息无法找到 Resource not found.**

**422 请求的body内容无效 The request body is not valid.**

### 示例

```
PUT /v1/delivery/{deliveryReferenceId}...
```

<https://partners-staging.baitime.cn/v1/delivery/{deliveryReferenceId}/status>

Content type: application/json

Request Payload Sample:

```
{
  "deliveryStatus": "REQUESTED",
  "estimatedDeliveryTime": "2007-12-03T10:15:30+01:00",
  "estimatedPickupTime": "2007-12-03T10:15:30+01:00",
  "courier": {
    "name": "Jane Doe",
    "phone": "+1-555-555-5555",
    "phoneCode": "111 11 111",
    "email": "email@email.com",
    "personalIdentifiers": {
      "taxIdentificationNumber": 1234567890
    }
  },
  "location": {
    "latitude": 38.8977,
    "longitude": 77.0365
  },
  "createdAt": "2007-12-03T10:15:30+01:00"
}
```

Status: 400/401

```
{
  "message": "The request body is invalid."
  "details": "The error validation details."
}
```

## 发布配送回调错误 **Publishes a delivery callback error.**

AUTHORIZATIONS: [OAuth2.0](#) (callback.error.write)

### Header参数: HEADER PARAMETERS

X-Store-Id string  
**Required**  
(必填)  
Example: 1  
The ID of the store in your system this call acts on behalf of.

---

X-Event-Id string  
**Required**  
(必填)  
Example: cf0ce51b-d74e-40d3-b177-1925ab4edc0c  
Unique identifier of the event that this callback refers to.

---

### Request Body模式 **REQUEST BODY SCHEMA: application/json**

errorCode string  
Enum: "CANCELLED" "UNKNOWN" "INVALID\_ARGUMENT"  
"FAILED\_PRECONDITION" "DEADLINE\_EXCEEDED" "NOT\_FOUND"  
"PERMISSION\_DENIED" "ALREADY\_EXISTS" "RESOURCE\_EXHAUSTED"  
"ABORTED" "OUT\_OF\_RANGE" "UNIMPLEMENTED" "INTERNAL"  
"UNAVAILABLE" "DATA\_LOSS" "UNAUTHENTICATED"  
Errors that occur when processing the webhook, modeled after Google's gRPC error codes.

---

errorMessage string or null  
Additional information about the error.

---

## 返回 Responses

**200** 回调错误被成功接受 **The callback error was successfully accepted.**

**400** 请求格式错误 **The request is malformed.**

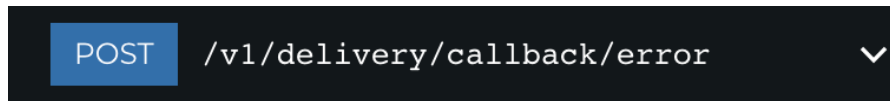
**401** 无效的验证信息 **Invalid authorization.**

**403** 未授权 **Authorization not valid for requested resource.**

**404** 请求的信息无法找到 **Resource not found.**

**422** 请求的body内容无效 **The request body is not valid.**

示例



<https://partners-staging.baitime.cn/v1/delivery/callback/error>

Content type: application/json

Request Payload Sample:

```
{
  "errorCode": "CANCELLED",
  "errorMessage": "string"
}
```

Status: 400/401

```
{
  "message": "The request body is invalid."
  "details": "The error validation details."
}
```

# 数据模型

## 订单 Order

externalIdentifiers    object (OrderExternalIdentifiers)

---

Required (必填)	订单的外部标识 ( 比如在第三方系统中的订单ID ) The identifiers of the order.
items Required (必填)	Array of objects (Item) 订单菜品 Items ordered.
orderedAt	string <date-time> 顾客下单时间 ( 以UTC时区为准 ) The date <b>time</b> (in UTC) when the order was placed by the customer.
currencyCode Required (必填)	string 3 characters 3位货币代码 ( 中国使用 : CNY ) The 3-letter currency code (ISO 4217) to use for all monetary values in this order.
customer	object (Person) 顾客信息 The delivery courier.
customerNote	string 顾客订单备注 ( 举例 : 不要加香菜 ) An order-level note provided by the customer.
status Required (必填)	string 订单状态。包含 : 新订单 , 订单确认 , 骑手已取餐 , 取消 , 已送达 , 备餐完成 , Enum: "NEW_ORDER" "CONFIRMED" "PICKED_UP" "CANCELED" "FULFILLED" "PREPARED" "UNKNOWN" The status of the order.
deliveryInfo	object (DeliveryInfo) 订单配送进度信息 Information on order's delivery process.
orderTotal Required (必填)	object (OrderTotal) 订单总金额 Details about values of the order.
customerPayments	Array of objects (CustomerPayment) 顾客付款信息 Details about the payments made by the customer.
fulfillmentInfo	object (FulfillmentInfo) 订单交付的信息 Information on order fulfillment.

```
{
  "externalIdentifiers": {
    "source": "ubereats",
    "id": "69f60a06-c335-46d9-b5a1-97f1a211c514",
    "friendlyId": "ABCDE"
  },
  "items": [
    {
      "id": "33e0418f-3d56-4360-ba03-18fc5f8844a3",
      "name": "Juicy Cheeseburger",
      "quantity": 1,
      "note": "Please cook to well done!",
      "categoryId": "303de078-870d-4349-928b-946869d4d69b",
      "categoryName": "Burgers",
      "price": 5.9,
      "modifiers": [
        {
          "id": "d7a21692-9195-43aa-a58f-5395bba8a804",
          "name": "Avocado",
          "quantity": 1,
          "price": 1,
          "groupName": "Add ons",
          "groupId": "fb52b138-7ac4-42c1-bfd8-664d57113a41",
          "modifiers": [
            {}
          ]
        }
      ]
    }
  ],
  "orderedAt": "2020-12-20T02:30:00Z",
  "currencyCode": "EUR",
  "customer": {
    "name": "Jane Doe",
    "phone": "+1-555-555-5555",
    "email": "email@email.com",
    "personalIdentifiers": {
      "taxIdentificationNumber": "01234567890"
    }
  },
  "customerNote": "Please include extra napkins!",
  "status": "NEW_ORDER",
  "deliveryInfo": {
    "courier": {
      "name": "Jane Doe",
      "phone": "+1-555-555-5555",
      "email": "email@email.com",
      "personalIdentifiers": {
        "taxIdentificationNumber": "01234567890"
      }
    }
  }
}
```

```
},
"destination": {
  "postalCode": "20500",
  "city": "Washington",
  "state": "DC",
  "countryCode": "US",
  "addressLines": "1600 Pennsylvania Avenue NW",
  "location": {
    "latitude": 38.8977,
    "longitude": 77.0365
  }
},
"licensePlate": "ABC 123",
"makeModel": "Honda CR-V",
"lastKnownLocation": {
  "latitude": 38.8977,
  "longitude": 77.0365
},
"note": "Gate code 123"
},
"orderTotal": {
  "subtotal": 11.97,
  "discount": 1,
  "tax": 1.1,
  "tip": 2,
  "deliveryFee": 5,
  "total": 19.07,
  "couponCode": "VWXYZ98765"
},
"customerPayments": [
  {
    "value": 2,
    "processingStatus": "COLLECTABLE",
    "paymentMethod": "CASH"
  }
],
"fulfillmentInfo": {
  "pickupTime": "2020-12-20T02:30:00Z",
  "deliveryTime": "2020-12-20T02:30:00Z",
  "fulfillmentMode": "DELIVERY",
  "schedulingType": "ASAP",
  "courierStatus": "COURIER_ASSIGNED"
}
}
```

## 菜单 **Menus**

menus	<p>店铺的所有菜单，每个菜单拥有唯一ID</p> <p>object (All Menus for the store, indexed by ID)</p> <p><b>店铺的菜单</b></p> <p>A menu for a store.</p>
items Required (必填)	<p>店铺的所有菜品，每个菜品拥有唯一ID</p> <p>object (All Items for the store, indexed by ID)</p> <p>菜单上售卖的菜品。菜品可以有不同的规格属性（称为“ModifierGroup”）规格同时也可以包含菜品，具体示例可参见“modifierGroups”的描述</p> <p>Items are sold on the Menu. Modifiers to items are items themselves and their relationship is defined by ModifierGroup.</p>
photos Required (必填)	<p>店铺所有的菜品图片，每张图片拥有唯一ID</p> <p>object (All ItemPhotos for the store, indexed by ID)</p> <p><b>呈现菜品照片需要的所有信息</b></p> <p>All data required to represent a Photo for an Item.</p>
categories Required (必填)	<p>店铺所有的菜品品类，每个品类拥有唯一ID</p> <p>object (All Categories for the store, indexed by ID)</p> <p><b>品类名称</b></p> <p>Categories contain a list of Item references.</p>
modifierGroups Required (必填)	<p>店铺所有的菜品规格，每个菜品规格拥有唯一ID</p> <p>object (All ModifierGroups for the Store, indexed by ID)</p> <p>菜品规格举例（比如：奶茶的杯型就是一个规格，其中包含：小杯、中杯、超大杯；另外奶茶的甜度是一个规格的例子，其中包含：100%，75%，50% 25%，无糖）</p> <p>Items are sold on the menu. Modifiers to items are items themselves and their relationship is defined by ModifierGroup.</p>

```
{
  "menus": {
    "1c753dac-5b48-11eb-ae93-0242ac130002": {
      "name": "Tasty BBQ",
      "description": "Cooking up BBQ deliciousness from around the globe!",
      "id": "ff6dd693-5e55-4a92-a359-ea61b23ed423",
      "categoryIds": [
        "string"
      ],
      "hours": {
        "intervals": [
```



```
{
  "day": "Monday",
  "fromHour": 8,
  "fromMinute": 0,
  "toHour": 20,
  "toMinute": 45
}
]
}
},
"items": {
  "2c75405e-5b48-11eb-ae93-0242ac130002": {
    "id": "faa4c79f-480d-4de1-bc34-5fb74ef082ef",
    "name": "Bagel",
    "description": "Delicious bagel!",
    "price": {
      "currencyCode": "EUR",
      "amount": 25.21
    },
    "priceOverrides": "$3.50 for UberEats, $4 for DoorDash.",
    "status": {
      "saleStatus": "TEMPORARILY_NOT_FOR_SALE",
      "suspendedUntil": "2020-11-23T21:33:51Z"
    },
    "modifierGroupIds": [
      "string"
    ],
    "photoIds": [
      "string"
    ],
    "photoUrls": [
      "string"
    ]
  }
},
"photos": {
  "3c754180-5b48-11eb-ae93-0242ac130002": {
    "id": "e9174f75-a293-4908-bba7-9db69871ad81",
    "contentType": "PNG",
    "url":
https://www.google.com/images/branding/googlelogo/2x/googlelogo\_color\_272x92dp.png
  }
},
"categories": {
  "4c7542c0-5b48-11eb-ae93-0242ac130002": {
    "name": "Breakfast",
    "description": "All early morning items served up hot and fresh!",
    "id": "da0e4e94-5670-4175-897a-3b7dde45bed5",
    "itemIds": [
      "string"
    ]
  }
}
```

```
}
},
"modifierGroups": {
  "5c7543b0-5b48-11eb-ae93-0242ac130002": {
    "id": "db5740de-429c-4f06-b264-3d41dce6d8c3",
    "name": "Choose your type of bagel",
    "description": "Choose any of these delicious types of bagels",
    "minimumSelections": 1,
    "maximumSelections": 1,
    "itemIds": [
      "string"
    ]
  }
}
}
```

## 错误信息 **Error Message**

message	string 错误信息描述 The error description.
---------	--

---

```
{
  "message": "The request body is invalid."
}
```

# 回调模式 **Webhook Models**

## 事件消息 **Event Notification**

eventId	string <uuid> 事件的唯一标识 Unique identifier of the event.
---------	---

---

eventTime	string <date-time> 事件发生的日期 Date of event occurrence.
eventType	string 事件类型 The type of the event.
metadata	object (MetadataObject) 事件的基本信息 Information about the event.

```
{
  "eventId": "d6703cc8-9e79-415d-ac03-a4dc7f6ab43c",
  "eventTime": "2019-08-24T14:15:22Z",
  "eventType": "string",
  "metadata": {
    "storeId": "string",
    "applicationId": "string",
    "resourceId": "string",
    "payload": {},
    "resourceHref": "string"
  }
}
```

## 订单 Order

externalIdentifiers <b>required</b>	object (OrderExternalIdentifiers) 订单的外部标识 ( 比如在第三方系统中的订单ID ) The identifiers of the order.
items <b>required</b>	Array of objects (Item) 订单的菜品 Items ordered.
orderedAt	string <date-time> 顾客下单的时间 ( 以UTC时区为准 )

	The date (in UTC) when the order was placed by the customer.
currencyCode required	string 3 characters 3位货币代码 ( 中国使用 : CNY ) The 3-letter currency code (ISO 4217) to use for all monetary values in this order.
customer	object (Person) 顾客信息 The delivery courier.
customerNote	string 顾客订单备注 ( 举例 : 不要加香菜 ) An order-level note provided by the customer.
status required	string 订单状态。包含 : 新订单 , 订单确认 , 骑手已取餐 , 取消 , 已送达 , 已完成 , 未知 Enum: "NEW_ORDER" "CONFIRMED" "PICKED_UP" "CANCELED" "FULFILLED" "PREPARED" "UNKNOWN" The status of the order.
deliveryInfo	object (DeliveryInfo) 订单配送信息 Information on order's delivery process.
orderTotal required	object (OrderTotal) 订单总金额 Details about values of the order.
customerPayments	Array of objects (CustomerPayment) 顾客付款信息 Details about the payments made by the customer.
fulfillmentInfo	object (FulfillmentInfo) 订单交付的信息 Information on order fulfillment.

```
{
  "externalIdentifiers": {
```

```
"source": "ubereats",
"id": "69f60a06-c335-46d9-b5a1-97f1a211c514",
"friendlyId": "ABCDE"
},
"items": [
  {
    "id": "33e0418f-3d56-4360-ba03-18fc5f8844a3",
    "name": "Juicy Cheeseburger",
    "quantity": 1,
    "note": "Please cook to well done!",
    "categoryId": "303de078-870d-4349-928b-946869d4d69b",
    "categoryName": "Burgers",
    "price": 5.9,
    "modifiers": [
      {
        "id": "d7a21692-9195-43aa-a58f-5395bba8a804",
        "name": "Avocado",
        "quantity": 1,
        "price": 1,
        "groupName": "Add ons",
        "groupId": "fb52b138-7ac4-42c1-bfd8-664d57113a41",
        "modifiers": [
          {}
        ]
      }
    ]
  }
],
"orderedAt": "2020-12-20T02:30:00Z",
"currencyCode": "EUR",
"customer": {
  "name": "Jane Doe",
  "phone": "+1-555-555-5555",
  "email": "email@email.com",
  "personalIdentifiers": {
    "taxIdentificationNumber": "01234567890"
  }
},
"customerNote": "Please include extra napkins!",
"status": "NEW_ORDER",
"deliveryInfo": {
  "courier": {
    "name": "Jane Doe",
    "phone": "+1-555-555-5555",
    "email": "email@email.com",
    "personalIdentifiers": {
      "taxIdentificationNumber": "01234567890"
    }
  }
},
"destination": {
  "postalCode": "20500",
  "city": "Washington",
```

```

"state": "DC",
"countryCode": "US",
"addressLines": "1600 Pennsylvania Avenue NW",
"location": {
  "latitude": 38.8977,
  "longitude": 77.0365
}
},
"licensePlate": "ABC 123",
"makeModel": "Honda CR-V",
"lastKnownLocation": {
  "latitude": 38.8977,
  "longitude": 77.0365
},
"note": "Gate code 123"
},
"orderTotal": {
  "subtotal": 11.97,
  "discount": 1,
  "tax": 1.1,
  "tip": 2,
  "deliveryFee": 5,
  "total": 19.07,
  "couponCode": "VWXYZ98765"
},
"customerPayments": [
  {
    "value": 2,
    "processingStatus": "COLLECTABLE",
    "paymentMethod": "CASH"
  }
],
"fulfillmentInfo": {
  "pickupTime": "2020-12-20T02:30:00Z",
  "deliveryTime": "2020-12-20T02:30:00Z",
  "fulfillmentMode": "DELIVERY",
  "schedulingType": "ASAP",
  "courierStatus": "COURIER_ASSIGNED"
}
}

```

## 订单状态历史 **Order Status History**

orderStatusHistory    Array of objects (OrderStatusEvent)  
 订单状态变化的列表  
 Array of OrderStatusEvent

---

```
{
  "orderStatusHistory": [
    {
      "status": "ORDER_ACCEPTED",
      "eventTime": "2019-08-24T14:15:22Z"
    }
  ]
}
```