

# Uber Eats

## API連携開発のご案内 1.2.1

関係者外秘

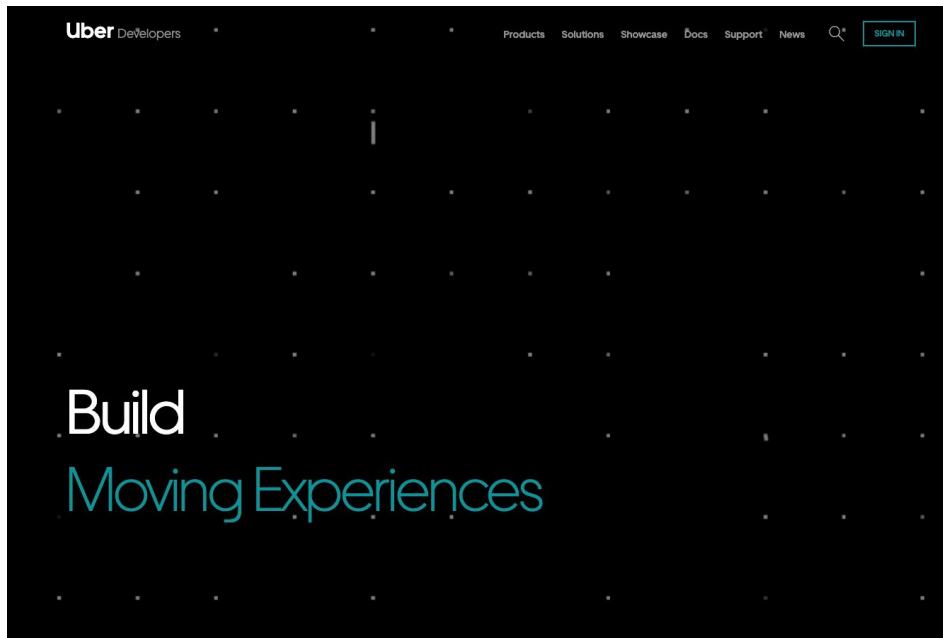
## Agenda

- 01** 開発アカウントの作成とアプリケーションの設定
- 02** テスト環境利用案内
- 03** API機能一覧
- 04** 開発の始めに
- 05** Order 連携処理フロー
- 06** Order連携についての参考情報・ヒント
- 07** メニュー連携及び参考情報・ヒント
- 08** リンク集

# 01 開発アカウントの作成とアプリケーションの設定

# 開発アカウントの作成とアプリケーションの設定

<https://developer.uber.com/> にアクセスし、右上の 'SIGN IN' ボタンをクリックし、画面に従いアカウントを作成してください。  
詳細手順は別途添付します。



# 02 テスト環境利用 案内

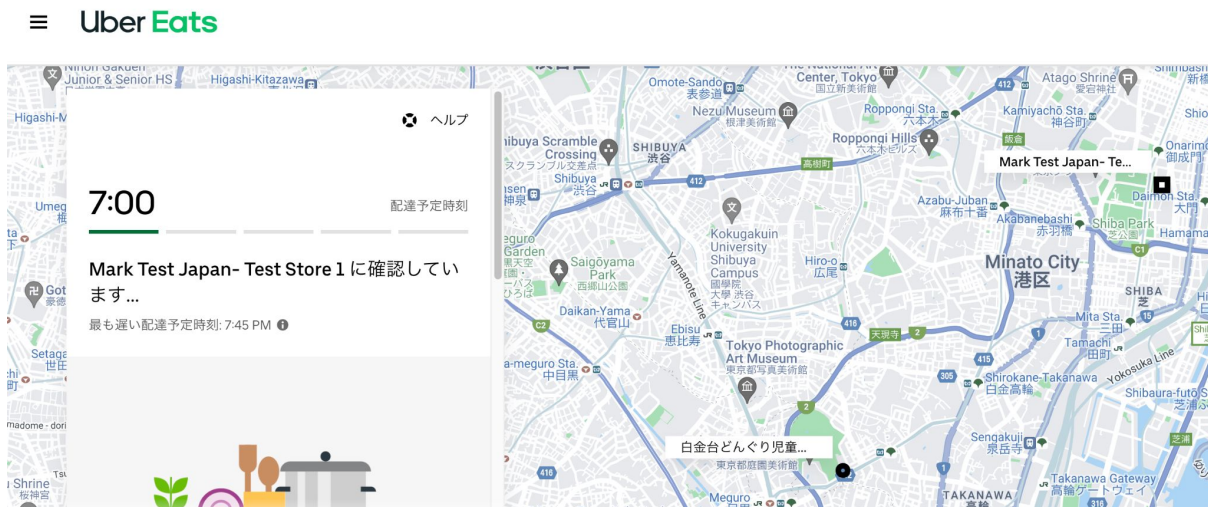
# 注文のテストについて

1. <https://www.ubereats.com/jp> (UberではテストURLと商用URLは同じもの)にアクセスして、**Test Store Account**でサインインしてください。(Login with passwordを選択)
2. 配達先の住所を設定してください。(テストストアの住所の周辺に設定してください)
3. Test Storeをクリックして、注文を行ってください。



# 注文のテストについて

4. 注文を行うと、下記のような状態になります。本資料の [05 Order API 連携の処理フロー]を参照して、webhook受信を確認し、Get Order Details API、Accept/Cancel/Deny Order APIのテストを行ってください。



# 03 API機能一覧



## Uber Eats API 機能一覧と実装要否

種別	機能	リンク	実装要否
Order Webhook	Order Notification webhook	<a href="https://developer.uber.com/docs/eats/api/webhook-orders-notification">https://developer.uber.com/docs/eats/api/webhook-orders-notification</a> "event_type": "orders.notification"	必須
	Order Cancelled webhook	<a href="https://developer.uber.com/docs/eats/api/webhook-orders-cancel">https://developer.uber.com/docs/eats/api/webhook-orders-cancel</a> "event_type": "orders.cancel"	マーチャントによる
Order	Get Order Details API	<a href="https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid">https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid</a>	必須
	Accept Order API	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-acceptposorder">https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-acceptposorder</a>	必須
	Cancel Order API	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-cancel">https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-cancel</a>	必須
	Patch Cart API	<a href="https://developer.uber.com/docs/eats/api/v2/patch-eats-orders-orderid-cart">https://developer.uber.com/docs/eats/api/v2/patch-eats-orders-orderid-cart</a>	小売店の場合は必須

種別	機能	リンク	実装要否
Order	Deny Order API	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-denyporder">https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-denyporder</a>	マーチャントによる
	Cancel Order APIのreason項目	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-cancel">https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-cancel</a> Request Parameters: reason(OUT_OF_ITEMS,RESTAURANT_TOO_BUSY) <b>reasonをOTHERを設定するとき、detailsに詳細原因を設定してください</b>	必須
	Deny Order APIのreason項目	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-denyporder">https://developer.uber.com/docs/eats/api/v1/post-eats-orders-orderid-denyporder</a> <b>Request Parameters: reason.explanationに詳細原因を設定してください</b>	Deny Order API 実装時は必須
	Get Order Details APIのプロモーション項目	<a href="https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid#response-body-promotion">https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid#response-body-promotion</a> Response Body: Promotions	必須

種別	機能	リンク	実装要否
Order	Get Order Details APIのカトラリー類のチェックボックス項目	<a href="https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid#response-body-disposable-items">https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid#response-body-disposable-items</a> Response Body: Disposable Items - should_include	マーチャントによる

種別	機能	リンク	実装要否
Menu	Get Menu API	<a href="https://developer.uber.com/docs/eats/api/v2/get-eats-stores-storeid-menus">https://developer.uber.com/docs/eats/api/v2/get-eats-stores-storeid-menus</a>	必須
	Upload Menu API	<a href="https://developer.uber.com/docs/eats/api/v2/put-eats-stores-storeid-menus">https://developer.uber.com/docs/eats/api/v2/put-eats-stores-storeid-menus</a>	必須
	Update Item API	<a href="https://developer.uber.com/docs/eats/api/v2/post-eats-stores-storeid-menus-items-itemid">https://developer.uber.com/docs/eats/api/v2/post-eats-stores-storeid-menus-items-itemid</a>	必須

種別	機能	リンク	実装要否
Integration Config APIs	OAuth Flow handled by POS Provider	<a href="https://developer.uber.com/docs/riders/guides/authentication/user-access-token">https://developer.uber.com/docs/riders/guides/authentication/user-access-token</a>	必須
	List All Stores API (with User Access Token)	<a href="https://developer.uber.com/docs/eats/api/v1/get-eats-stores">https://developer.uber.com/docs/eats/api/v1/get-eats-stores</a>	必須
	Get Integration Details API	<a href="https://developer.uber.com/docs/eats/api/v1/get-eats-stores-storeid-posdata">https://developer.uber.com/docs/eats/api/v1/get-eats-stores-storeid-posdata</a>	必須
	Activate Integration API	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-stores-storeid-posdata">https://developer.uber.com/docs/eats/api/v1/post-eats-stores-storeid-posdata</a>	必須
	Remove Integration API	<a href="https://developer.uber.com/docs/eats/api/v1/delete-eats-stores-storeid-posdata">https://developer.uber.com/docs/eats/api/v1/delete-eats-stores-storeid-posdata</a>	必須
	Update Integration Details API	<a href="https://developer.uber.com/docs/eats/api/v1/patch-eats-stores-storeid-posdata">https://developer.uber.com/docs/eats/api/v1/patch-eats-stores-storeid-posdata</a>	必須

種別	機能	リンク	実装要否
Integration Config APIs	Integration Updated Webhook	<a href="https://developer.uber.com/docs/eats/api/webhook-store-provisioned">https://developer.uber.com/docs/eats/api/webhook-store-provisioned</a>	必須
	Integration Disabled Webhook	<a href="https://developer.uber.com/docs/eats/api/webhook-store-deprovisioned">https://developer.uber.com/docs/eats/api/webhook-store-deprovisioned</a>	必須

種別	機能	リンク	実装要否
Store	Get Store Details API	<a href="https://developer.uber.com/docs/eats/api/v1/get-eats-stores-storeid">https://developer.uber.com/docs/eats/api/v1/get-eats-stores-storeid</a>	推奨
	List All Stores API(with Client Credentials Token)	<a href="https://developer.uber.com/docs/eats/api/v1/get-eats-stores">https://developer.uber.com/docs/eats/api/v1/get-eats-stores</a>	推奨
	Get Restaurant Status API	<a href="https://developer.uber.com/docs/eats/api/v1/get-eats-store-storeid-status">https://developer.uber.com/docs/eats/api/v1/get-eats-store-storeid-status</a>	推奨
	Set Restaurant Status API	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-store-storeid-status">https://developer.uber.com/docs/eats/api/v1/post-eats-store-storeid-status</a>	推奨
	Get Store Holiday Hours API	<a href="https://developer.uber.com/docs/eats/api/v1/get-eats-stores-storeid-holidayhours">https://developer.uber.com/docs/eats/api/v1/get-eats-stores-storeid-holidayhours</a>	推奨
	Set Store Holiday Hours API	<a href="https://developer.uber.com/docs/eats/api/v1/post-eats-stores-storeid-holidayhours">https://developer.uber.com/docs/eats/api/v1/post-eats-stores-storeid-holidayhours</a>	推奨

# 04 開発の始めに



# Integration Config APIの実装について

店舗の連携設定をするために、最初に Integration Configの実装が必要です。

Integration Config APIでは、他のAPIが使うClient Credentials Tokenだけではなく、User Access Tokenの取得も必要です。

<https://developer.uber.com/docs/eats/guides/integration-activation-flows>

と<https://developer.uber.com/docs/riders/guides/authentication/user-access-token>

をご確認ください。

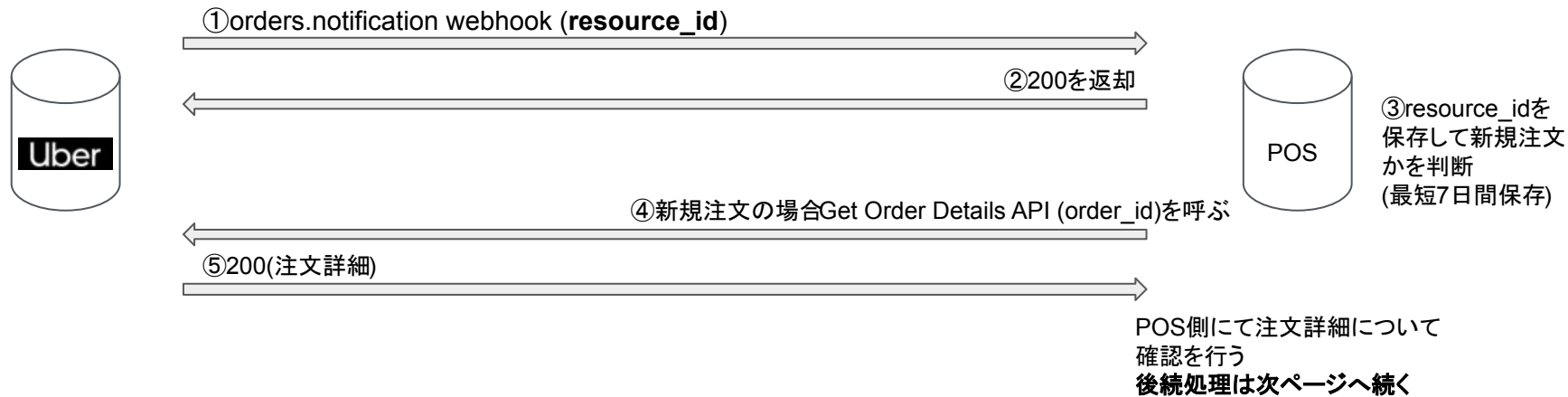
※1: テスト環境にてUser Access Token取得時の認証は**Test UEM Account**を使ってください

※2: **Webhook URL**を設定していない場合、Integration Config APIを実行できない(実行エラーになる)ので、ページ6に従ってWebhook URLを設定してください。

※3: Activate Integration APIを実行する際、body部に**"is\_order\_manager": true**を入れてください。

# 05 Order 連携処 理フロー

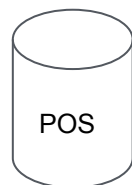
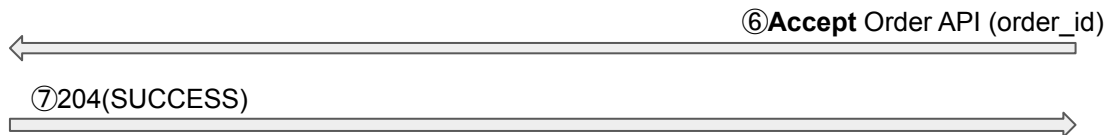
# Order API 連携の処理フロー



- ※ webhookに対して速やかに200を返却してください。(500,502-504を返却する場合や返却しない場合、webhookが再送(retry)されます。それ以外のコードを返却する場合は再送しません。POS側にてロードバランサーなどを経由する場合も、200で返却するよう注意してください)。
- ※ インターネットの一時的な遅延でwebhookが再送される場合もありますので、**200を返却した後、POS側の後続処理にて新規注文か再送注文かについて、resource\_idで判断するロジックを必ず実装してください。**  
webhookのresource\_idとOrder APIのorder\_idは同じもので、個別注文を特定するidです(一意性を保っています)。  
詳細: <https://developer.uber.com/docs/eats/introduction>

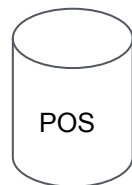
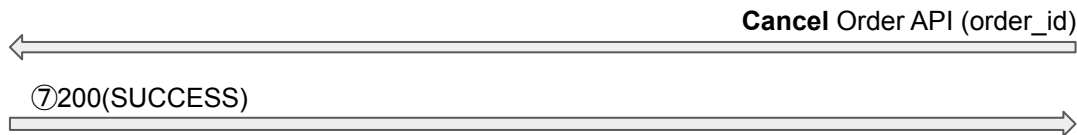
# Order API 連携の処理フロー

前ページから続く。  
当該注文を受け付ける場合

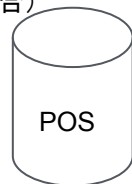
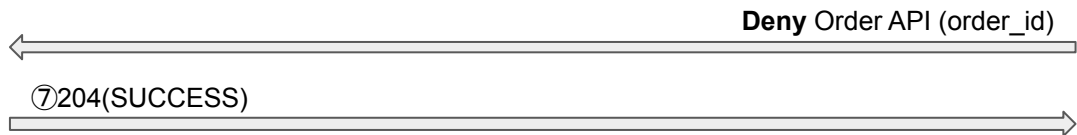


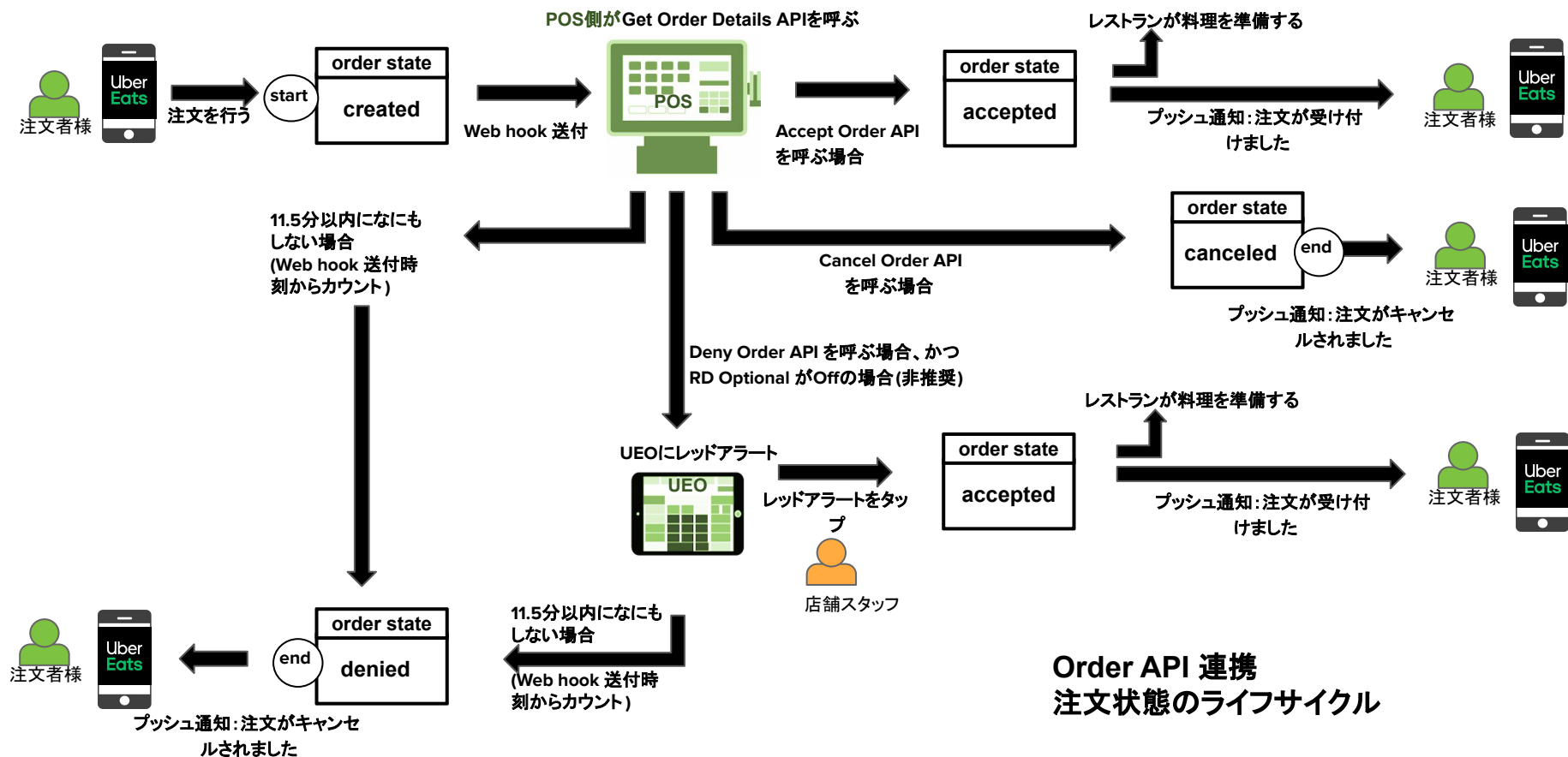
※前ページのwebhook受信後、11.5分以内にAccept Order APIを呼ばないと注文が自動キャンセルされます。

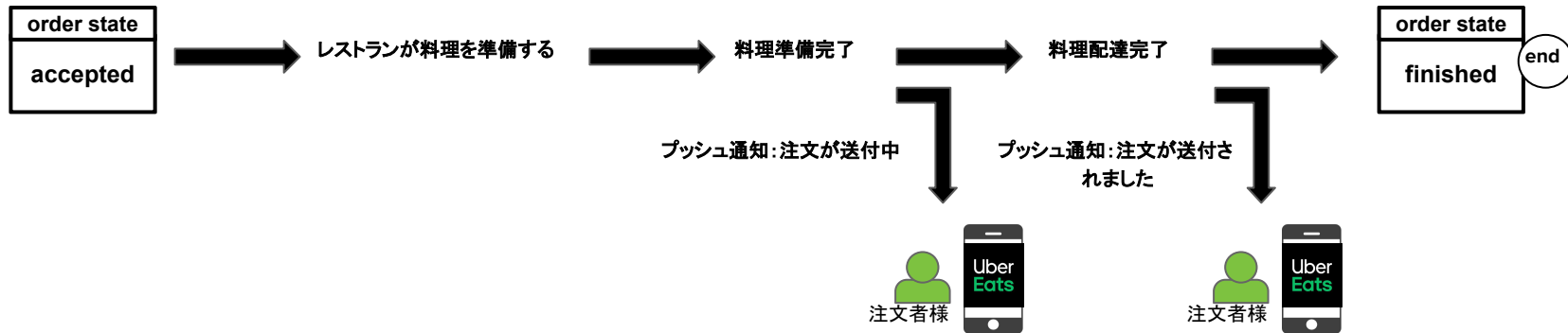
⑥あるいは、  
当該注文を受け付けない場合(注文の中の一部商品が品切れの場合や、レストランが忙しすぎて注文を用意できない場合)



⑥あるいは、  
当該注文にPOS側システムに登録されていない商品が存在する場合。  
(POS側システムに商品登録漏れや商品dを誤登録した場合)







注文をaccepted以降のフローはAPI POS連携前と同じ  
(当ページはAPI 連携と直接関係ありません)

# 06 Order連携につ いての参考情報・ヒ ント

# HTTPレスポンス エラーハンドリング

ライフサイクルを終えた注文(ページ23-24に"end"が付いているorder\_state)に対して、Accept/Deny Order APIを呼ぶと、404エラー("message": "Order not found")が返却されます。Cancel Order APIを呼ぶと、404エラー("message": "not found")が返却されます。エラーハンドリングを実装してください

(POS側がAccept/Cancel/Deny Order APIを呼ぶ前に注文者様が自らキャンセルするケースもあるため)。

stateはGet Order Details APIで取得できます。(Get Order Details APIで取得するjsonのrootのcurrent\_stateがorder\_stateになります。)

7日以内の注文に対して、ライフサイクルを終えてもGet Order Details APIにて情報を取得できます。

詳細: <https://developer.uber.com/docs/eats/guides/order-states>

<https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid>



# HTTPリクエスト リトライ

下記の場合、HTTPリクエスト(APIを呼ぶ)を間隔を空けて数回リトライしてください(例えば一回目のリトライは10秒後に実施、二回目のリトライは20秒後に実施、三回目のリトライを30秒後に実施)。

1. 7日以内の注文に対してGet Order Details APIが404エラーを返却した場合。
2. APIを呼んだ後、想定外のエラーコードを返却される場合(例えば500エラー)。
3. APIを呼んだ後、15秒経過してもHTTPレスポンスが返ってこない場合。(Upload Menu APIの場合は60秒)

# Tokenの使用について (Client Credentials Tokenと User Access Token)

取得したTokenは、有効期限内 (30日)に何回も使えるので POS側システムのサーバー側に保管してください。

Tokenの重複取得はUber側のサーバーに負担をかけるのであるべくしないでください。

頻繁にTokenを取得する場合、一時的にToken取得が制限される場合があります。

右の図のresponseのexpires\_inが有効期限になります。(2592000秒=30日)

詳細: <https://developer.uber.com/docs/eats/guides/authentication>

POST

https://login.uber.com/oauth/v2/token

Params

Authorization

Headers (8)

Body

Pre-request Script

none

form-data

x-www-form-urlencoded

raw

binary

	KEY	VALUE
<input checked="" type="checkbox"/>	client_id	{{uber_api_client_id}}
<input checked="" type="checkbox"/>	client_secret	{{uber_api_client_secret}}
<input checked="" type="checkbox"/>	grant_type	client_credentials
<input checked="" type="checkbox"/>	scope	eats.order

Body

Cookies

Headers (13)

Test Results (2/2)

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "access_token": "JA.
    VUNmGAAAAAAAEgASAAAABwAIAwAAAAAAAEgAAAAAAAHQA
    AAEAAAAD_12_B8wi5W99ZuQn-AgCuAAAAA2nxcwtp1Rj1vbT6j
    VMd8Z65FGWTB720hMNs9u90D1NvoTRbAjGaR1340AHQ4o-kok!
    9w7MUewc0MAAAABjFvuJycuxZYGyV6JAAAAGIwZDg10DAzLTM:
3    "token_type": "Bearer",
4    "expires_in": 2592000,
5    "scope": "eats.order"
6  }

```

## Order API 連携 POS側システムとの店舗のマッピング

POS側システムとの店舗のマッピングは Order APIのstoreのid(UUID)をご利用ください

POS側システムにて当該 storeのidはどの店舗を指しているのかのマッピング情報を用意してください。

注文web hookがPOS側システムに送信されるとき、どの店舗に対する注文なのかは、店舗のマッピング情報で判断してください。

webhookのuser\_idとOrder APIのstoreのidは同じもので、個別店舗を特定する idです(一意性を保っています)。

```
"meta": {  
  "resource_id": "153dd7f1-339d-4619-940c-418943c14636",  
  "status": "pos",  
  "user_id": "89dd9741-66b5-4bb4-b216-a813f3b21b4f"  
},
```

詳細: <https://developer.uber.com/docs/eats/api/v2/get-eats-order-orderid>  
<https://developer.uber.com/docs/eats/api/webhook-orders-notification>

# Order API 連携 プロモーション項目に関する実装

Get Order Details APIの中のプロモーション項目に関して、

最低でも下記の promo\_type、promo\_discount\_value、discount\_items項目の取得の実装が必要になります。

promo\_discount\_valueは当該注文の合計割引値になります。( Get Order Details APIに商品ごとの割引値は現在対応しておりません。)

promo\_discount\_percentage項目につきましては、現在日本では適用していません。(値は常に 0か1)

```
"promotions": {  
  "promotions": [  
    {  
      "external_promotion_id": "",  
      "promo_type": "BOGO",  
      "promo_discount_value": -49000,  
      "promo_discount_percentage": 0,  
      "discount_items": [  
        {  
          "external_id": "アップルジュース",  
          "discounted_quantity": 1  
        },  
        {  
          "external_id": "サラダ",  
          "discounted_quantity": 1  
        }  
      ]  
    }  
  ]  
}
```

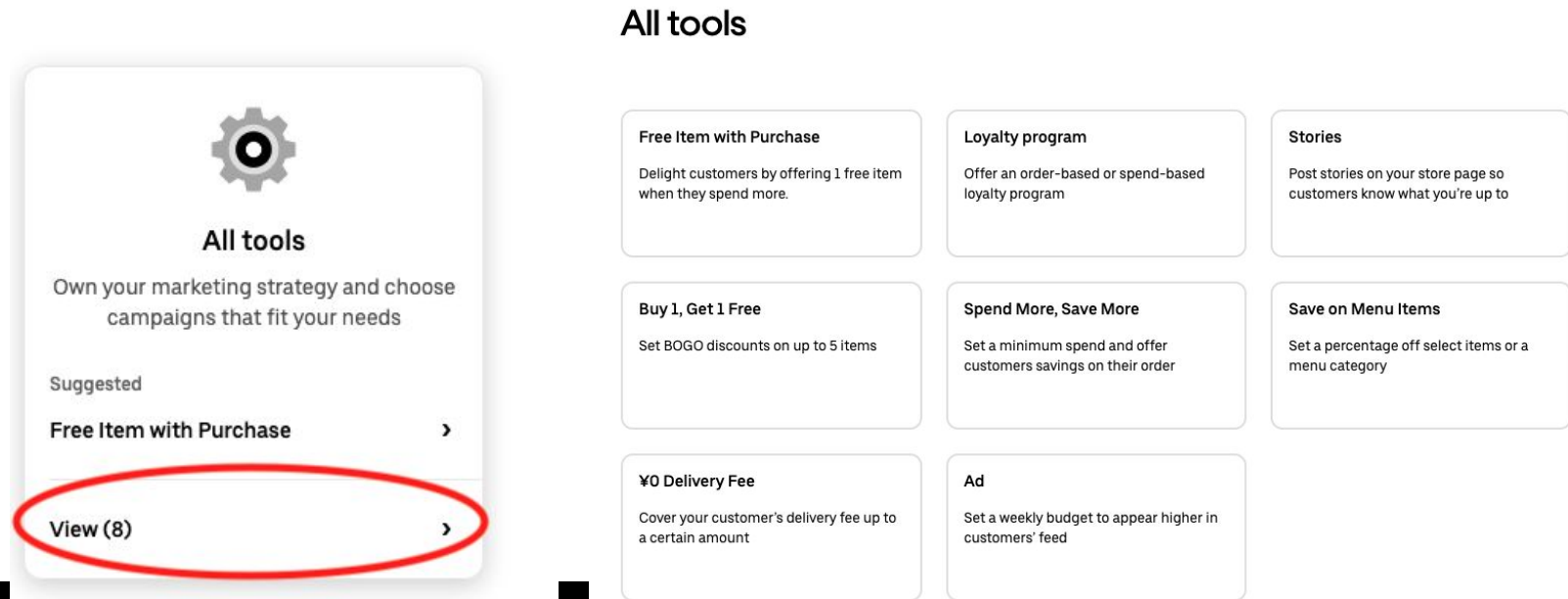
## 各種プロモーション(割引キャンペーン)の設定方法(設定方法はAPI連携実施前と同じ)

1. ブラウザを開いて UEM (UberEatsManager): <https://merchants.ubereats.com/manager> にアクセスし、**テストアカウント** でサインインしてください。
2. “Marketing” のタブをクリックして、“Pick a promotion” をクリックしてください。

The screenshot displays the Uber Eats Manager web interface. On the left is a vertical sidebar with navigation links: Home, Orders, Stores, Analytics, Insights, Feedback, Top Eats, Reports, Webshop, Payments, Holiday Hours, and Marketing. The 'Marketing' link at the bottom is highlighted with a red circle. The main content area is titled 'Promotions' and includes a sub-note: 'Data will update every 48 hours.' Below this, there is a promotional banner with an illustration of a person holding a drink, the text 'Give customers even more reasons to order, from item savings to free delivery.', and a button labeled 'Pick a promotion' which is also circled in red. Above the banner, there is another section with the text 'Advertise your store and reach more customers. Start a campaign in just a few minute:' and a 'Create an ad' button.

# 各種プロモーション(割引キャンペーン)の設定方法(設定方法はAPI連携実施前と同じ)

3. “All tools”の”View”をクリックすると、各種プロモーションが表示されます



# 各種プロモーション(割引キャンペーン)の設定方法(設定方法はAPI連携実施前と同じ)

4. 実施したいプロモーションを選んでください。

種類1 Buy 1, Get 1 Free (promo\_type:BOGO): 1つ頼むと1つ無料

種類2 ¥0 Delivery Fee (promo\_type:FREEDELIVERY): 無料配送

種類3 Spend More, Save More (promo\_type:FLATOFF): 指定される注文金額で自動割引

種類4 Free Item with Purchase (promo\_type:FREEITEM\_MINBASKET): 無料アイテム

種類5 Save on menu items (promo\_type:MENU\_ITEM\_DISCOUNTあるいはCATEGORY\_DISCOUNT): アイテム割引

5. プロモーション実施後、<https://www.ubereats.com/jp/>にて注文を行うと、Get Order Details APIにてプロモーションに関する注文情報( "promo\_type"などの項目)を取得できます。(ページ 30を参考)

# Order API 連携 カトラリー類のチェックボックス項目に関する実装

Get Order Details APIの中のカトラリー類のチェックボックス項目に関して、disposable\_itemsのshould\_includeがtrueの場合、カトラリーを要求されています。

```
"packaging": {  
  "disposable_items": {  
    "should_include": true  
  }  
},
```

カトラリー類、ストローなどのご要望

☐

レストランへのメモを追加してください

お会計に進む・¥1,080



## POS側システム商品(アイテム)とのマッピング

注文に対してPOS側システム商品とのマッピングは Get Order Details APIのResponseのcartのitemsのid項目をご利用ください。

※External Dataを利用してマッピングする案もありましたが、非推奨になっています。どうしても External Dataでマッピングする必要がある場合、ご相談ください。

itemsのid項目の編集/更新はUpload Menu APIを使います。(例えば”id”: “A23456789012345”で更新する)

まずGet Menu APIで既存メニューのjson取得して、そのjsonの中のitemsのidを編集/変更して、Upload Menu APIを使ってください。

<https://developer.uber.com/docs/eats/api/v2/put-eats-stores-storeid-menus>

# 07 メニュー連携及 び参考情報・ヒント

# Menu API 連携

## メニュー構造

メニュー構造はレストラン様のメニューの特徴によって設計してください。

次の2ページは参考テンプレートになりますが、必ずしもレストラン様のメニュー構造に最適しているわけではありません。

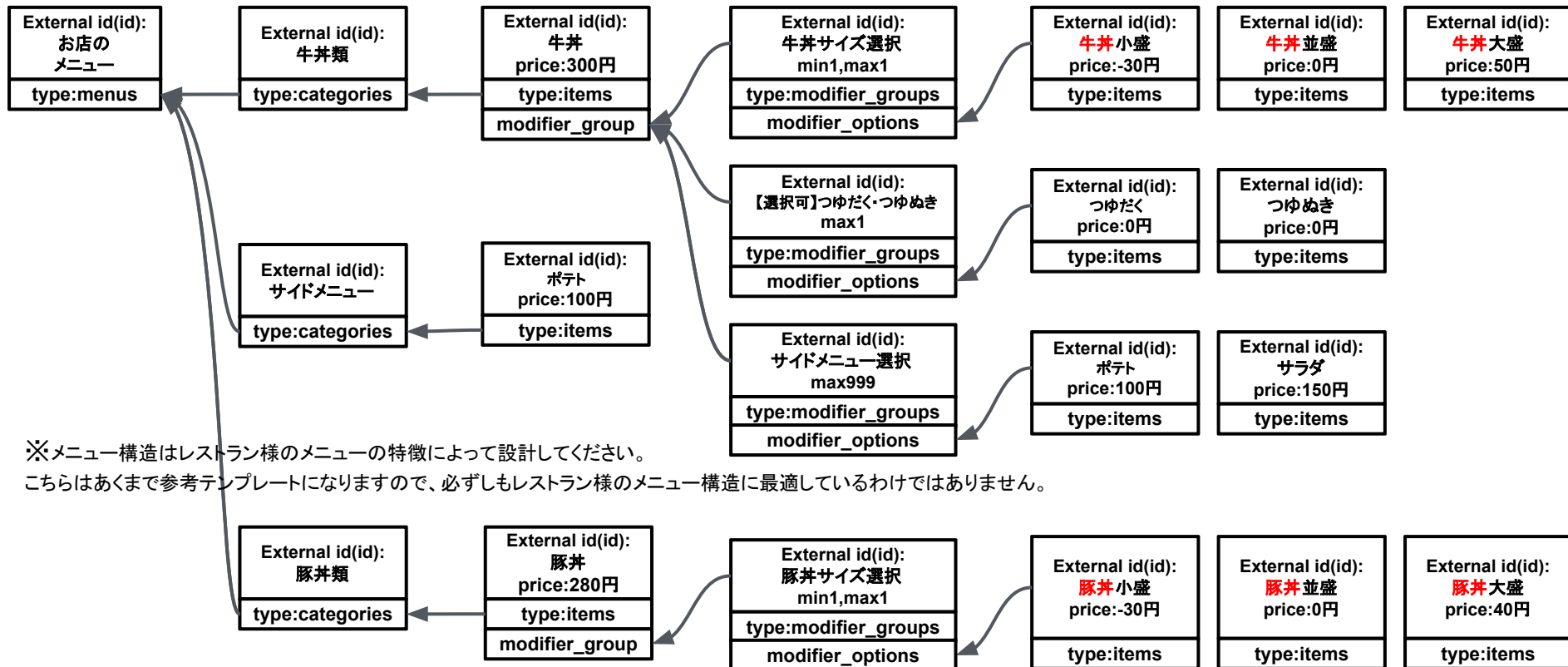
**現時点のメニュー構造 (json)はテスト店舗にて Get Menu APIを使って取得して確認してください。**

(API開発途中、商用店舗からテスト店舗へのメニュー反映は Uber側に依頼してください。)

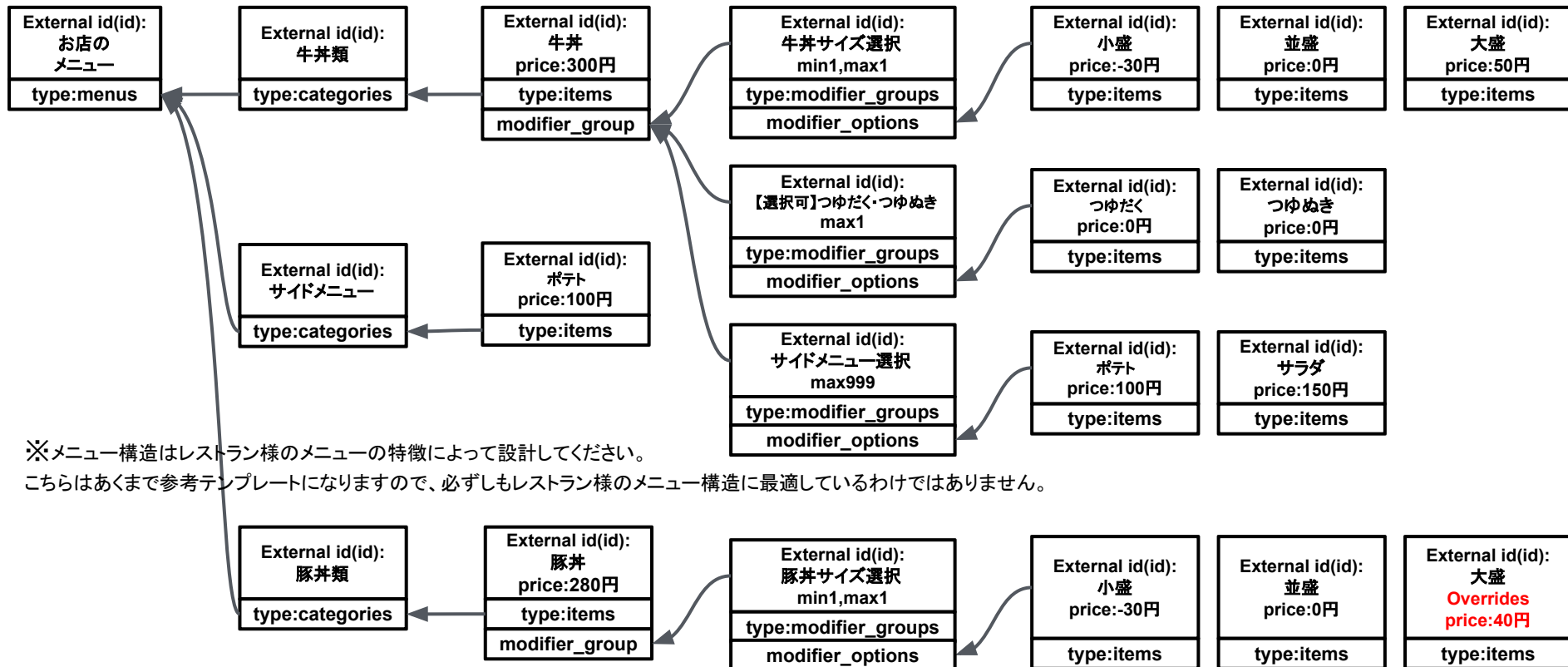
まずは現時点のメニュー構造 (json)の確認から始めてください。

<https://developer.uber.com/docs/eats/api/v2/get-eats-stores-storeid-menus>

## 価格のoverrideを使わないパターン(一般的なパターン)



## 価格のoverrideを使うパターン



※メニュー構造はレストラン様のメニューの特徴によって設計してください。

こちらはあくまで参考テンプレートになりますので、必ずしもレストラン様のメニュー構造に最適しているわけではありません。

# Menu API 連携

## Upload Menu API のjsonの保管について

Uber側はUpload Menu API のjsonのログを保管しません。  
メニュー更新に関する問い合わせや調査を依頼する際、Upload Menu API のjsonのログが必要になりますので  
御社側のシステムにて少なくとも7世代保管するようおすすめします。

## 商品の売り切れの設定や価格変更の設定

リアルタイムでの個別商品の売り切れ(**suspend\_until**)の設定や価格変更の設定を行う場合はUpdate Item APIをご利用ください。

<https://developer.uber.com/docs/eats/api/v2/post-eats-stores-storeid-menus-items-itemid>

### Request Body Parameters - Suspension

**Describes why, and until when, an item is suspended from sale.**

Name	Type	Description
suspend_until	int	(optional) The time at which the item will return to being available for sale, specified as a Unix timestamp in seconds since Jan 1, 1970. A null value, or time in the past, indicates that an item is available - otherwise it will be shown as “Sold Out” and unavailable to order.

## 商品の売り切れの設定や価格変更の設定

夜間にて全商品一括(全商品を1API CALL)で商品ごとの売り切れ設定(**suspend\_until**)や売り切れ設定解除、価格変更を行う場合はUpload Menu APIをご利用ください。

(1jsonの中で全商品を商品(id)単位で設定可能)

<https://developer.uber.com/docs/eats/api/v2/put-eats-stores-storeid-menus>

まずGet Menu APIを呼んでjson bodyを取得して、  
それをベースに内容を更新してUpload Menu APIを呼んでください。

```
{
  "id": "4719853583467",
  "title": {
    "translations": {
      "en": "ポテトSサイズ"
    }
  },
  "description": {
    "translations": {
      "en": "Sサイズのポテトです。"
    }
  },
  "image_url": "https://aaa.bbb.jpeg",
  "price_info": {
    "price": 4000,
    "overrides": []
  },
  "quantity_info": {
    "quantity": {},
    "overrides": []
  },
  "suspension_info": {
    "suspension": {
      "suspend_until": 8640000000
    }
  }
}
```

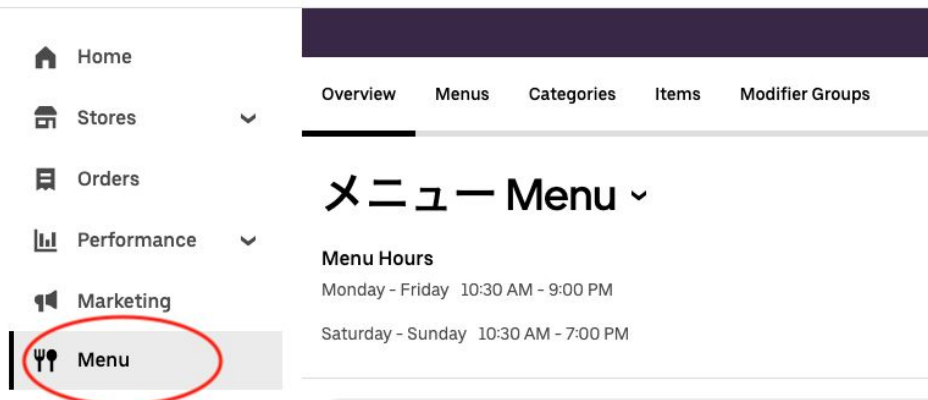


# Menu API 連携

## Upload Menu API 使用する場合の注意事項

Upload Menu APIを使用する場合、Uber Eats ManagerのMenuタブ (Menu Maker)でメニューの更新することをお控えください。メニュー不整合が発生する可能性があるため、Upload Menu APIとMenu Makerを混ざってのご利用はお勧めしないことになっています。Upload Menu APIをご利用の場合、メニューの更新(メニュー追加、商品追加、商品の属性変更など)はすべてUpload Menu APIで行ってください。

### Uber Eats Manager



# 08 リンク集

# リンク集

Product Name	説明	Link	Login Account(テスト環境)
API Documentation	API仕様書	<a href="https://developer.uber.com/docs/eats/introduction">https://developer.uber.com/docs/eats/introduction</a>	N/A
Developer Dashboard	開発アプリケーション設定のサイト	<a href="https://developer.uber.com/">https://developer.uber.com/</a>	Developer Account
Web Uber Eats	Uber EatsのWebバージョン。注文を行う時に使います。	<a href="https://www.ubereats.com">https://www.ubereats.com</a>	Test Store Account
Uber Eats Manager	店舗管理画面。店舗関連の設定、プロモーション設定に使います。	<a href="https://restaurant.uber.com/">https://restaurant.uber.com/</a>	Test UEM Account
Menu Maker	メニュー編集のUIツール。Menu APIとの混用は不可	<a href="https://restaurant.uber.com/manager/menu-maker/店舗のstoreUUID/overview">https://restaurant.uber.com/manager/menu-maker/店舗のstoreUUID/overview</a>	Test UEM Account
Uber Eats Orders	店舗用の注文受付UberタブレットのWebバージョン。	<a href="https://merchants.ubereats.com/orders/dashboard_v2/orders/all">https://merchants.ubereats.com/orders/dashboard_v2/orders/all</a>	Test Store Account
Postman Bundle	Uber Eats APIのPostman Collections と Environment。	<b>Uber Eats APIs Postman Bundle</b>	Client ID/Secret

Uber Eats