Introduction to Eats APIs

> **Access to These APIs May Require Written Approval From Uber**
>
> Uber's APIs are always under development and as such are subject to changes according to our Versioning & Upgrade policy. As part of Uber's ongoing privacy improvements, we've updated our Developer API program with new scope access policies for third party applications. For further information, please refer to our **Getting Started** guides.

The Eats APIs enable partners to programmatically manage stores, menus and orders on the Uber Eats platform.

Overview

The **Integration Activation API** can be used to associate your application to a merchant's location within Uber. Using the merchant's authorization, you can retrieve their store-list, and allow them to select the locations to configure and activate.

Uber Developers

overrides). These APIs can also be used to manage app access and integration details on a store level.

The **Menu API** provides the ability to manage menus as well as regular store hours.

Ingest and manage incoming orders from Uber Eats with the **Order API**. The API includes webhooks for incoming order notifications and cancellations from Uber Eats. Retrieve, accept, deny, and cancel orders with the Order API endpoints.

Use the **Reporting API** to request and access standardized reporting. The API includes a notification webhook once a report is completed.

Integration Activation API Suite

API Endpoints

- `GET Stores`
  Retrieve the locations for a given user authorized to your application.
- `POST Activate Integration`
  Associate application to a merchant's location.
- `GET Integration Data`
  Retrieve specific integration information for a location.

- `PATCH Update Integration Data`

  Update integration configuration for a location.

- `DELETE Remove Integration`

  Remove your application association from a store.

Webhook Notifications

- `store.provisioned Webhook`

  Inform application a location has been provisioned to your application.

- `store.deprovisioned Webhook`

  Inform application a location has been de-provisioned from your application.

Menu API Suite

API Endpoints

- `GET Menu`

  Retrieve existing store menu.

- `PUT Menu`

  Create, update, delete menu and menu items.

- `POST Update Item`

  Update item availability and pricing.

Store Management
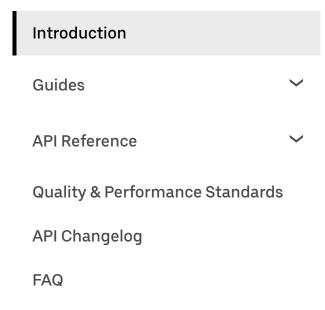
API Endpoints

- `GET Store Details`

  Retrieve store information.
- `GET List All Stores`

  Retrieve all stores provisioned to developer.
- `GET/POST Restaurant Status`

  Retrieve store availability status and set stores on/offline.
- `GET POS Status`

  Check if stores are order-integrated.
- `POST POS Data`

  Activate store integration.
- `DELETE POS Data`

  Remove store integration.
- `PATCH Update POS Status`

  Disable or enable order integration.

- `GET/POST Holiday Hours`
  Retrieve and set date-specific store hours.

Order Workflow Management

API Endpoints

- `orders.notification Webhook`
  Get Notified when an order is placed on a location your application is enabled.
- `GET Order Details`
  Retrieve order details.
- `POST Order Acceptance`
  Accept order.
- `POST Order Denial`
  Deny order.
- `POST Cancel Order`
  Cancel accepted orders.
- `POST Resolve Fulfillment Issues`
  Resolve fulfillment issues due to cart issues
- `Mark Order Ready`
  Inform Uber an order is ready to assist in modeling dispatch of

delivery partners.

- `POST Adjust Order Price`
  Adjust the order price based on customer contacting restaurant.
- `POST Update Ready Time`
  Update the expected time for an order to be ready for pickup.
- `GET Orders`
  Retrieve orders.

Webhook Notifications

- `orders.failed Webhook`
  Notification event for cancelled Eats order.

- `order.fulfillment_issues.resolved Webhook`
  When a customer has adjusted their cart based on usage of Order Fulfillment endpoint.

Reporting API Suite

API Endpoints

Navigation (left sidebar):

Introduction

Guses ⌄

API Reference ⌄

Quality & Performance Standards

API Changelog

FAQ

- `POST Report`
  Request reporting.

Webhook Notifications

- `eats.report.success Webhook`
  Notification event for completed report job.

Point-of-Sale (POS) Integrations

Get the most value out of the APIs by setting them up to synchronize your store menu from your store's POS system.

Once this is set up, you can inject Eats orders straight to the POS. This eliminates the need for manual and duplicate processes between Eats and the POS, such as marking items as sold out or transposing order details from tablets.

Next steps

Before starting an integration, we first suggest reviewing the following page(s):

- Versioning & Upgrade Timelines
  To ensure you can accomodate our upgrade schedules and that your

To ensure your integration is built in a way that can meet the standards and performance criteria outlined in this document.

Once you are ready, visit our **Getting Started** page to begin the integration process.