

# GIGABLE SCALES WITH AWS AND HYPERTRACK BY AUTOMATING WORKFLOWS FOR ON-DEMAND WORKFORCE

Many organizations with highly mobile field service teams, with supply chain and logistics requirements, and on-demand labor build their own workflow applications to help automate their mobile worker order planning, assignment, dispatch, and tracking. One of such companies is [Gigable](#), a leading all-in-one staffing for short-term staffing. Gigable has partnered with [HyperTrack](#) to leverage the [AWS platform](#) via HyperTrack use case driven APIs to add live location driven workflow automation to their own platform.

In this post, we will discuss how [Amazon Location Service](#) (ALS) and other AWS managed service offerings are integrated in the HyperTrack platform to provide the power of API driven integrations to Gigable. HyperTrack is an [AWS](#) Supply Chain Competency Partner and AWS Marketplace Seller that is a leading provider of building blocks to automate on-demand jobs and workforce workflows. [HyperTrack cloud](#) enables customers to deliver automated workflow solutions for companies building their own workforce management applications and platforms.

## AMAZON LOCATION SERVICE OVERVIEW

Amazon Location Service is a fully managed service that makes it easier for developers to add location functionality like geocoding, routing, geofencing, and mapping to their applications. AWS has made it generally available in December 2020 for all AWS customers. HyperTrack embraced ALS routing solution as a part of its core API platform offering to implement routing and ETA functionality for its platform customers. With customer centric use case driven APIs, such as [Orders](#), [Drivers](#), and [Places APIs](#) HyperTrack empowers its customers to automate their applications and platforms with the power of live location based route planning, assignment, and tracking.



## HYPERTRACK OVERVIEW

As an [AWS Supply Chain Competency Partner](#) and [AWS Marketplace Seller](#) HyperTrack delivers public APIs to enable critical capabilities such as order assignment and route optimization, live location tracking, improving fulfillment rates, and much more for its customers. HyperTrack customers are able to get off the ground to incorporate these capabilities within a few weeks of integration work and testing.

HyperTrack delivers APIs to support these use cases and more:

- Live order tracking
- Dispatching orders to nearby drivers
- Order assignment and route optimization
- Measure productivity of visits and routes
- Driver payments based on ground truth
- Visibility enablement for operations teams

HyperTrack built APIs for these use cases above leveraging years of experience of understanding of these use cases with stellar technology execution while leveraging AWS at scale, with [Amazon Location Service](#) at the core of its route planning and tracking implementation. HyperTrack next generation [mobile SDK](#) offers a unique blend of configurability, battery drain management, real-timeness and accuracy. This mobile SDK is integrated at the core of customer applications and provides real-time tracking component and location, activity, and status data streaming necessary to generate ground truth for workers on the move. To ensure visibility of workers on the move, HyperTrack has implemented features that allow customers not only effectively collect ground truth from mobile devices during working hours and shifts, but also provide exceptional reporting ability when location tracking ability is impacted by user and system level outages in these devices. HyperTrack provides unsurpassed outage reporting capabilities to its developers which can, in turn, create powerful fleet management solutions for their customers.



## INTEGRATION OVERVIEW

This article addresses the solution that Gigable implemented with HyperTrack APIs to capitalize on HyperTrack and Amazon Location Service to create part time staffing platform solution.



As described in the diagram above, Gigable uses HyperTrack Mobile SDK, HyperTrack APIs, and [HyperTrack Embeddable Dashboards](#) and [Views](#) to incorporate routing and live ETAs for on-demand gig worker workflow operations.

HyperTrack Mobile SDK is integrated in Gigable app and enables live location tracking as described in the diagram in Step 1 above. Mobile SDK performs tracking for the Gigable worker only during work shift hours.

In Step 2 above, HyperTrack platform leverages a number of AWS services, such as AWS Lambda, API Gateway, Kinesis, DynamoDB with streams, and many others to ingest and process location streaming data, performing real time location accuracy processing, summarizing a wide array of activities data, user and system driven outages, and creating a wealth of history and analytics data.

To help manage mobile worker workflows, HyperTrack provides [Orders API](#) as shown in Step 3 in the diagram. Orders API conveniently abstracts order workflow management. It initiates tracking on the worker's device, manages creation of real-time route summary information as well as ETA prediction with the help of Amazon Location Service routes capability. The worker's activity tracked since the start of the shift, with the destination geofence visit being captured in the worker history timeline. Gigable platform completes the worker's order via another Orders API call. The completion of the order in turn generates the order history summary, and stops HyperTrack Mobile SDK from tracking the worker's device location unless there are any other orders currently assigned and dispatched to the worker's device.

In order to provide compelling real-time worker fleet visibility experiences, HyperTrack utilizes AWS AppSync with AWS Lambda to implement real-time embeddable views and dashboards as shown in Step 4.

To monitor and execute on worker order activity, HyperTrack provides webhook payloads to convey status of the order while it is being tracked as shown in Step 5 above.

As shown in Step 6, Gigable uses HyperTrack Orders API to create, dispatch, and complete orders for its workers on the move. In Step 7, Gigable embeds HyperTrack embeddable views and dashboards to provide real-time order tracking experiences to its customers.

To achieve scale and performance of HyperTrack APIs, HyperTrack has spent multiple man years of engineering design and development while leveraging the infrastructure of AWS services, such as AWS Lambda, Kinesis, DynamoDB streams, and many others, with Amazon Location Service being at the core of route planning and tracking.

# WALKTHROUGH

## STEP 1: INTEGRATE HYPERTRACK MOBILE SDK

Integrate HyperTrack Mobile SDK inside your mobile app. Please refer to [iOS](#) and [Android](#) SDK documentation on how to proceed.

To automatically start order tracking you must integrate silent push notifications for [iOS](#) and [Android](#). The device will stop tracking when all active routes for device are completed. HyperTrack uses a combination of silent push notifications and sync method on the SDK to ensure that tracking starts and stops on the device as expected.



## PREREQUISITES

- Create an account with **AWS**.
- You can use this account to get started with **HyperTrack in AWS Marketplace**.
- Create a **HyperTrack** account.
- **Get started with HyperTrack APIs!**

## STEP 2: CALL HYPERTRACK ORDERS API TO CREATE AN ORDER

The HyperTrack Orders API lets you track live orders with ETAs, delays, arrivals, risks and more through an API call. Each API call will result in the following:

- Automatically starting driver tracking on the device when dispatched and stops tracking upon route completion
- Generated order completion markers that are used to generate distances traveled per order and deviations from destination
- Creation of a mobile-friendly share URL per order for customers to get a real-time order tracking experience
- Provision of a dashboard-friendly embed URL per route for ops managers to get a real-time route view

Create on-time routes with a fixed sequence of orders by passing in the "track\_mode":"on\_time" flag to the [Orders API](#) request.

On-time routes additionally provide:

- Estimated ETAs with live traffic conditions and re-routes
- Webhook payload notifications about order delays and on time arrival scores when optional scheduled\_at attribute is used in the Orders API call to create an order

Order data is used to generate insights on route efficiency, on-time arrival, completion at destination, and time spent at destination. Past routes may be replayed for non-repudiation in case of escalations or other exceptions.

```
{
  "device_id": "00000000-C3C6-4678-89B5-259BDE4A9C26",
  "track_mode": "on_time",
  "orders": [{
    "order_handle": "test-order-01",
    "scheduled_after": "2023-03-22T22:00:00Z",
    "scheduled_at": "2023-03-22T23:00:00Z",
    "destination": {
      "geometry": {
        "type": "Point",
        "coordinates": [74.24219516188421, 16.693551259417756]
      },
      "address": "██████████, ",
      "radius": 50
    },
    "expected_service_time": 800
  }
]
}
```

To create and track a route with an order, you must define an Order API payload structured as follows:

Parameter `expected_service_time` specifies time in seconds that driver expects to service an order. Note that `order_handle` must be a unique identifier for your order, while `destination` is required and `scheduled_at` and `expected_service_time` parameters are optional.

```
var request = require('request');

let AccountId = 'account_id';
let SecretKey = 'secret_key';
let AUTHORIZATION_TOKEN = btoa(`${AccountId}:${SecretKey}`);

console.log(AUTHORIZATION_TOKEN)

body: JSON.stringify({
  "device_id": "00000000-BD4D-46A0-BED9-53BAF09A41B1",
  "track_mode": "on_time",
  "orders": [
    {
      "order_handle": "order-00001",
      "scheduled_after": "2023-04-05T22:00:00Z",
      "scheduled_at": "2023-04-05T23:00:00Z",
      "destination": {
        "geometry": {
          "type": "Point",
          "coordinates": [-122.398096, 37.793038]
        },
        "address": "██████████, San Francisco, CA 94108, United States",
        "radius": 50
      },
      "product_type": ["plumber"],
      "capacity_used": 1,
      "expected_service_time": 800,
      "type": "drop",
      "metadata": {"customerId": "2233322"}
    }
  ]
})

);
request(options,function (error, response) {
if (error)thrownew Error(error);
  console.log(response.body);
});
```

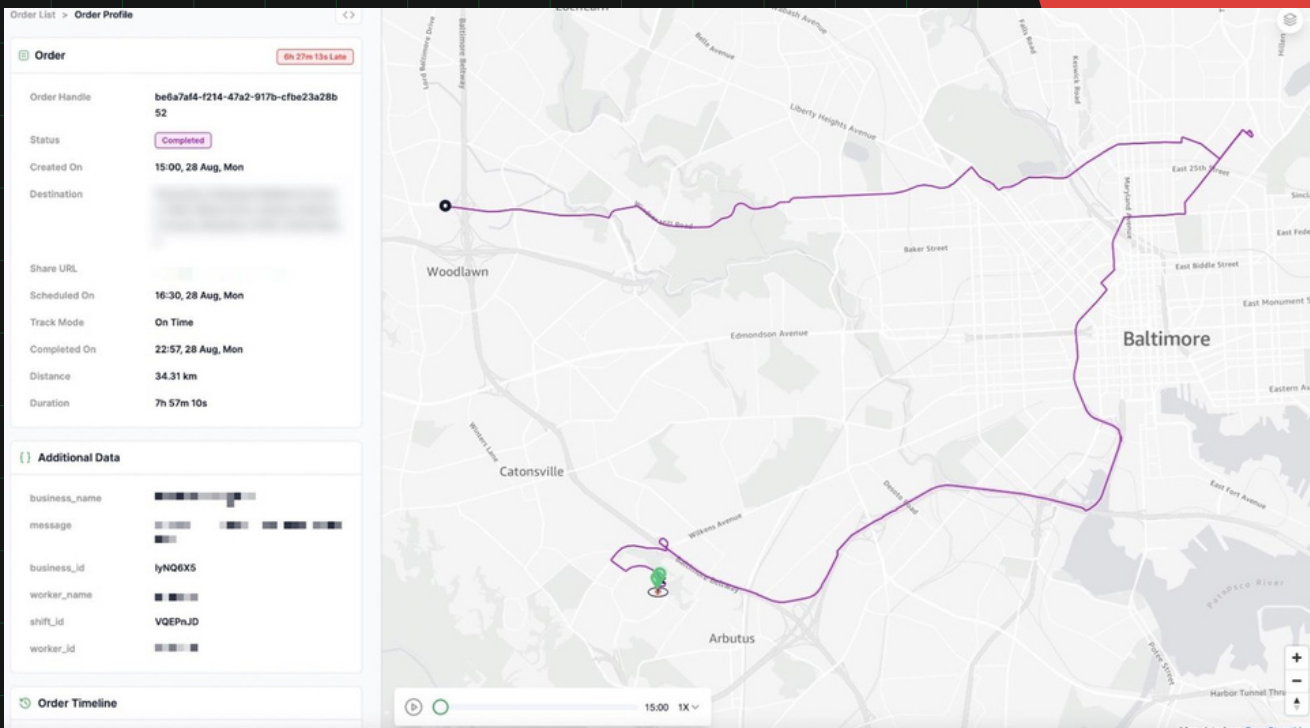


### STEP 3: EMBED ROUTE VIEWS IN OPS DASHBOARDS

Orders API returns `embed_url` upon successful creation of the route.

The `embed_url` is built for embedding in ops dashboards. While the route is ongoing, ops dashboard users will see a desktop-web view of the route's timeline, live location, arrival status, estimated route and ETA to destination.

For completed routes, dashboard will see the route summary with a detailed timeline from start to complete.



While shareable views are accessible to everyone with the URL, embed views includes your publishable key. Although the permalink of each URL in the dashboard includes the publishable key, your publishable key is available in the [Setup page](#) of your dashboard.

The embed URL format is [`<https://embed.hypertrack.com/trips/{route_handle}?publishable_key={publishable_key}>`]  
(`<https://embed.hypertrack.com/trips/%7Broute_handle%7D?publishable_key=%7Bpublishable_key%7D>`)

## STEP 4: GET ORDER STATUS, TRACK ORDER DELAYS AND NO-SHOWS WITH WEBHOOK PAYLOADS

For on-demand workers with time slot or end time commitments, it is important to proactively manage delays and potential no-shows. Delays might trigger an automatic workflow, notify operations managers or reset customer expectations.

Use `scheduled_at` property in [ISO8601 format](#) to the order destination to indicate a scheduled arrival. Use this when the order has a time commitment to the customer. When set, HyperTrack will send [order delay webhooks](#) when the driver is highly likely to miss that time commitment.

With `scheduled_at`, operations pre-emptively act on delays and catch potential no-shows before they escalate.

## STEP 5: COMPLETE THE ORDER AND GET ORDER SUMMARY

Use Complete Order API to complete the order and GET Orders APIs to retrieve active and completed route data.

For all active and completed routes, a summary object is returned as part of the route response payload. If routes is still in active state, the summary object will contain the latest from the driver's device.

## CONCLUSION

This article demonstrates how AWS customers can leverage HyperTrack APIs and extend the capabilities of AWS services such as Amazon Location Service to create, assign, and dispatch orders for on-demand workforce. AWS customers can integrate HyperTrack Mobile SDK in their apps, while their order management systems can seamlessly integrate with HyperTrack Orders API to manage order workflow. To add visibility for their operations managers and other users, HyperTrack customers can integrate embeddable views and dashboards. At the same time, HyperTrack customers can integrate webhook payloads to process and manage order status updates in real-time within their own order management systems, be it third party or home grown software deployments.