

FDE Technical Challenge - Outbound Carrier Campaign

Overview

You will implement a real-world use case for inbound carrier engagement using the HappyRobot platform. Your AI assistant will be able to receive calls from carriers looking to book loads.

Goals

Objective 1: Implement Inbound Use Case

Use the HappyRobot platform to create an inbound agent where the AI assistant gets calls from carriers.

The loads will be searched using an API in a file or DB which will contain the context within the following fields for each load:

Field	Description
load_id	Unique identifier for the load
origin	Starting location
destination	Delivery location
pickup_datetime	Date and time for pickup
delivery_datetime	Date and time for delivery

equipment_type	Type of equipment needed
loadboard_rate	Listed rate for the load
notes	Additional information
weight	Load weight
commodity_type	Type of goods
num_of_pieces	Number of items
miles	Distance to travel
dimensions	Size measurements

The assistant must:

- Get their MC number and verify they are eligible to work using the FMCSA API.
- Search the load and pitch the details.
- Ask if they're interested in accepting the load.
- If they make a counter offer evaluate it. Handle up to 3 back and forth's negotiating the offer.
- If a price is agreed, transfer the call to a sales rep.
- Extract from the call the most relevant data for the offer.
- Classify the call based on its outcome.
- Classify the sentiment of the carrier in the call.



Bonus Points

- Create a dashboard/report generation with use case metrics.
- Containerize the solution with Docker.

Deliverables

1. Link to your code repository.
2. Link to the configured outbound campaign in the HappyRobot platform.
3. A short video (5 mins) walking through:
 - Use case setup
 - Short demo
 - Optional dashboard if implemented

Additional Considerations

1. Security:

If you're creating an API, add basic security features such as:

- HTTPS (self-signed locally is fine, use Let's Encrypt or equivalent if deployed)
- API key authentication for all endpoints
- Bonus: use environment variables or secret managers to store sensitive values

2. Deployment:

- Deploy your API to a cloud provider of your choice (e.g., AWS, Google Cloud, Azure, [Fly.io](https://fly.io), etc.)
- Provide clear instructions on how to:
 - Access the deployment.
 - Reproduce your deployment if needed (e.g., Terraform, shell script, or manual steps)

3. Calls:

- **Do not buy a phone number on the platform. Use the web call feature.**