



i-Riide Company Fleet Management API Guide



Table of Contents

1. [Introduction](#)
 2. [Getting Started](#)
 3. [Authentication](#)
 4. [API Endpoints](#)
 5. [Examples](#)
 6. [Error Handling](#)
 7. [Best Practices](#)
 8. [Support](#)
-



Introduction

Welcome to the **i-Riide Company Fleet Management API**! This API allows your company to manage your entire fleet of vehicles, parking locations, EV charging stations, and micro-mobility devices through a secure, RESTful interface.

What You Can Do

- **Manage Cars:** Add, update, archive, and track your vehicle fleet
- **Manage Parking:** Create and manage parking locations
- **Manage EV Chargers:** Set up and monitor charging stations
- **Manage Micro-Mobility:** Handle bikes, scooters, and other devices
- **Analytics:** Get insights into your fleet performance
- **Secure Access:** All operations are protected with API key authentication

✓ Production Ready

- **Database Connected:** All operations are connected to live Firestore database
 - **Real-time Data:** Changes are immediately reflected in the database
 - **Secure Authentication:** API key validation against database records
 - **Unified Collections:** Same collections for users and companies (companies are users with `isCompany: true`)
-

Getting Started

Prerequisites

- Your company API key (contact support@i-riide.com to get one)
- Basic knowledge of HTTP requests
- Any programming language or tool that can make HTTP requests

Base URL

Production: `https://api-tl4e3rloia-uc.a.run.app`
Development: `http://localhost:5001/i-riide-prod/us-central1/api`

API Documentation

Interactive API documentation is available at:

`https://api-tl4e3rloia-uc.a.run.app/api-docs`

Authentication

API Key Format

Your API key must be included in all requests as a header:

```
X-API-Key: your-company-api-key-here
```

Authentication Endpoint

Before using the API, you can verify your API key:

```
POST /api/auth/company
```

Request:

```
{
  "apiKey": "your-company-api-key-here"
}
```

Response:

```
{
  "authenticated": true,
  "companyId": "your-company-id",
  "companyName": "Your Company Name",
  "message": "Authentication successful",
  "expiresAt": "2025-07-29T10:30:00.000Z",
  "apiKeyInfo": {
    "lastUsed": "2025-07-29T10:30:00.000Z",
    "expiresAt": "2025-01-15T10:30:00.000Z"
  }
}
```



API Endpoints



Car Fleet Management

Create a New Car

POST /api/companies/{companyId}/cars

Headers:

Content-Type: application/json
X-API-Key: your-company-api-key

Request Body:

```
{
  "title": "Tesla Model 3",
  "brand": "Tesla",
  "model": "Model 3",
  "description": "Electric sedan for city driving",
  "year": "2023",
  "licensePlate": "ABC-123",
  "seats": 5,
  "fuelType": "electric",
  "pricePerHour": 25.00,
  "pricePerDay": 150.00
}
```

Response:

```
{
  "id": "car123",
  "companyId": "your-company-id",
  "title": "Tesla Model 3",
  "brand": "Tesla",
  "model": "Model 3",
  "isAvailable": true,
  "isArchived": false,
  "createdAt": "2025-07-29T10:30:00.000Z",
  "message": "Car created by company"
}
```

Get All Cars

```
GET /api/companies/{companyId}/cars
```

Update a Car

```
PUT /api/companies/{companyId}/cars/{carId}
```

Archive a Car

```
PATCH /api/companies/{companyId}/cars/{carId}/archive
```

Unarchive a Car

```
PATCH /api/companies/{companyId}/cars/{carId}/unarchive
```

Parking Management

Create a Parking Location

```
POST /api/companies/{companyId}/parkings
```

Request Body:

```
{
  "title": "Downtown Parking Garage",
  "location": "123 Main St, Downtown",
  "description": "Secure parking garage in city center",
  "parkingSize": "large",
  "detailOption": "indoor",
  "capacity": 100,
  "pricePerHour": 5.00,
  "pricePerDay": 30.00
}
```

Get All Parking Locations

```
GET /api/companies/{companyId}/parkings
```

Update Parking Location

```
PUT /api/companies/{companyId}/parkings/{parkingId}
```

Archive/Unarchive Parking

```
PATCH /api/companies/{companyId}/parkings/{parkingId}/archive
PATCH /api/companies/{companyId}/parkings/{parkingId}/unarchive
```

EV Charger Management

Create an EV Charger

```
POST /api/companies/{companyId}/ev-charging
```

Request Body:

```
{
  "title": "Fast Charging Station",
  "location": "456 Electric Ave, Tech District",
  "description": "High-speed charging station",
  "chargingStationType": "type_2",
  "powerOutput": 50,
  "pricePerMinute": 0.30
}
```

Get All EV Chargers

```
GET /api/companies/{companyId}/ev-charging
```

Update/Archive EV Charger

```
PUT /api/companies/{companyId}/ev-charging/{chargerId}
PATCH /api/companies/{companyId}/ev-charging/{chargerId}/archive
PATCH /api/companies/{companyId}/ev-charging/{chargerId}/unarchive
```

Micro-Mobility Management

Create a Micro-Mobility Device

```
POST /api/companies/{companyId}/micro-mobility
```

Request Body:

```
{
  "title": "Electric Scooter #001",
  "mobilityType": "electric_scooter",
  "brand": "Xiaomi",
  "model": "Mi Electric Scooter",
  "location": "789 Mobility St, City Center",
  "maxSpeed": 25,
  "range": 45,
  "pricePerHour": 8.00,
  "pricePerDay": 40.00
}
```

Get All Micro-Mobility Devices

```
GET /api/companies/{companyId}/micro-mobility
```

Update/Archive Micro-Mobility Device

```
PUT /api/companies/{companyId}/micro-mobility/{mobilityId}
PATCH /api/companies/{companyId}/micro-mobility/{mobilityId}/archive
PATCH /api/companies/{companyId}/micro-mobility/{mobilityId}/unarchive
```

Analytics & Reporting

Fleet Overview

```
GET /api/companies/{companyId}/fleet/overview
```

Response:


```

{
  "companyId": "your-company-id",
  "fleetOverview": {
    "cars": {
      "total": 25,
      "active": 20,
      "archived": 5,
      "available": 15,
      "inUse": 5
    },
    "parking": {
      "total": 10,
      "active": 8,
      "archived": 2,
      "available": 6,
      "occupied": 2
    },
    "evChargers": {
      "total": 15,
      "active": 12,
      "archived": 3,
      "available": 8,
      "inUse": 4
    },
    "microMobility": {
      "total": 50,
      "active": 45,
      "archived": 5,
      "available": 30,
      "inUse": 15
    }
  },
  "lastUpdated": "2025-07-29T10:30:00.000Z"
}

```

Fleet Analytics

GET /api/companies/{companyId}/fleet/analytics?period=month

Response:

```
{
  "companyId": "your-company-id",
  "analytics": {
    "utilizationRate": 0.75,
    "revenuePerDay": 1250.50,
    "averageBookingDuration": 2.5,
    "popularServices": ["cars", "parking", "evChargers"],
    "peakHours": ["09:00", "17:00"],
    "monthlyGrowth": 0.15
  },
  "period": "month",
  "lastUpdated": "2025-07-29T10:30:00.000Z"
}
```



Examples

Complete Car Management Workflow

1. Create a Car

```
curl -X POST https://api-tl4e3rloia-uc.a.run.app/api/companies/your-company-id/cars \
-H "Content-Type: application/json" \
-H "X-API-Key: your-company-api-key" \
-d '{
  "title": "BMW X5",
  "brand": "BMW",
  "model": "X5",
  "year": "2023",
  "licensePlate": "XYZ-789",
  "seats": 7,
  "fuelType": "hybrid",
  "pricePerHour": 35.00,
  "pricePerDay": 200.00
}'
```

2. Get All Cars

```
curl -X GET https://api-tl4e3rloia-uc.a.run.app/api/companies/your-company-id/cars
-H "X-API-Key: your-company-api-key"
```

3. Update a Car

```
curl -X PUT https://api-tl4e3rloia-uc.a.run.app/api/companies/your-company-id/cars
-H "Content-Type: application/json" \
-H "X-API-Key: your-company-api-key" \
-d '{
  "pricePerHour": 40.00,
  "isAvailable": false
}'
```

4. Archive a Car

```
curl -X PATCH https://api-tl4e3rloia-uc.a.run.app/api/companies/your-company-id/cars
-H "X-API-Key: your-company-api-key"
```

JavaScript Example

```
const API_KEY = 'your-company-api-key';
const COMPANY_ID = 'your-company-id';
const BASE_URL = 'https://api-tl4e3rloia-uc.a.run.app/api';

// Create a car
async function createCar(carData) {
  const response = await fetch(`${BASE_URL}/companies/${COMPANY_ID}/cars`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'X-API-Key': API_KEY
    },
    body: JSON.stringify(carData)
  });

  return await response.json();
}

// Get fleet overview
async function getFleetOverview() {
  const response = await fetch(`${BASE_URL}/companies/${COMPANY_ID}/fleet/overview`, {
    headers: {
      'X-API-Key': API_KEY
    }
  });

  return await response.json();
}

// Usage
const newCar = await createCar({
  title: 'Tesla Model Y',
  brand: 'Tesla',
  model: 'Model Y',
  fuelType: 'electric',
  pricePerHour: 30.00
});
```

```
const overview = await getFleetOverview();  
console.log('Fleet Overview:', overview);
```

Python Example

```
import requests

API_KEY = 'your-company-api-key'
COMPANY_ID = 'your-company-id'
BASE_URL = 'https://api-tl4e3rloia-uc.a.run.app/api'

headers = {
    'X-API-Key': API_KEY,
    'Content-Type': 'application/json'
}

# Create a parking location
def create_parking(parking_data):
    response = requests.post(
        f'{BASE_URL}/companies/{COMPANY_ID}/parking',
        headers=headers,
        json=parking_data
    )
    return response.json()

# Get analytics
def get_analytics(period='month'):
    response = requests.get(
        f'{BASE_URL}/companies/{COMPANY_ID}/fleet/analytics',
        headers=headers,
        params={'period': period}
    )
    return response.json()

# Usage
parking = create_parking({
    'title': 'Airport Parking',
    'location': 'Airport Terminal 1',
    'capacity': 200,
    'pricePerHour': 8.00
})

analytics = get_analytics('month')
print(f"Monthly Revenue: ${analytics['analytics']['revenuePerDay']}")
```

Error Handling

Common Error Responses

400 Bad Request

```
{
  "error": "Validation Error",
  "message": "Invalid car data",
  "details": [
    {
      "type": "field",
      "value": "",
      "msg": "Car title must be 1-100 characters long",
      "path": "title"
    }
  ]
}
```

401 Unauthorized

```
{
  "error": "Unauthorized",
  "message": "Company API key required"
}
```

403 Forbidden

```
{
  "error": "Forbidden",
  "message": "Invalid or inactive company API key"
}
```

429 Too Many Requests

```
{
  "error": "Too Many Requests",
  "message": "Rate limit exceeded. Please try again later.",
  "retryAfter": 900
}
```

Rate Limiting

- **General API:** 100 requests per 15 minutes per IP
 - **Authentication:** 5 attempts per 15 minutes per IP
 - **Headers:** Check `X-RateLimit-Remaining` for remaining requests
-

Security Best Practices

API Key Security

1. **Keep your API key secret** - Never share it publicly
2. **Use HTTPS** - Always make requests over secure connections
3. **Rotate keys regularly** - Contact support to rotate your API key
4. **Monitor usage** - Check your API usage regularly

Data Validation

1. **Validate input data** - Ensure all required fields are provided
2. **Use appropriate data types** - Follow the API specifications
3. **Sanitize user input** - Remove any potentially harmful content

Error Handling

1. **Handle all error responses** - Don't ignore error messages
 2. **Implement retry logic** - For rate limiting and temporary errors
 3. **Log errors appropriately** - For debugging and monitoring
-

Support

Getting Help

- **Email:** support@i-riide.com
- **Documentation:** <https://api-tl4e3rloia-uc.a.run.app/api-docs>
- **Response Time:** 24 hours for general inquiries

Getting Your API Key

1. Contact support@i-riide.com
2. Provide your company information
3. Receive your secure API key
4. Start using the API immediately

API Status

- **Status Page:** Check API status at status.i-riide.com
- **Maintenance:** Scheduled maintenance notifications via email
- **Updates:** API updates and new features announced via email

Quick Reference

Base URL

```
https://api-tl4e3rloia-uc.a.run.app
```

Authentication Header

```
X-API-Key: your-company-api-key
```

Common Endpoints

- **Cars:** `/api/companies/{companyId}/cars`
- **Parking:** `/api/companies/{companyId}/parking`
- **EV Chargers:** `/api/companies/{companyId}/ev-chargers`
- **Micro-Mobility:** `/api/companies/{companyId}/micro-mobility`
- **Analytics:** `/api/companies/{companyId}/fleet/analytics`

Response Format

All responses are in JSON format with the following structure:

```
{
  "id": "resource-id",
  "companyId": "your-company-id",
  "message": "Operation successful",
  "data": { ... }
}
```

What's Next?

1. **Get your API key** from support@i-riide.com
2. **Test the API** using the interactive documentation
3. **Integrate with your systems** using the provided examples
4. **Monitor your fleet** using the analytics endpoints
5. **Scale your operations** as your fleet grows

i-Riide Company Fleet Management API

Empowering companies to manage their fleet efficiently and securely

Last Updated: August 6, 2025

Version: 1.0

For the latest updates, visit: <https://i-riide.com/api>