

REDfirm Ltd
Unit 4 - CBC
Eismann Way
Corby
NN17 5ZB

+44 (0)1536 737212

matt@redfirm.co.uk
www.redfirm.co.uk



FleetFusion

Product Overview

Routing & Scheduling API

Version 1.0

Document Type:

Product Overview

Document Issued:

29.03.2021

Contents

FleetFusion Routing API	3
System Overview	3
System Architecture	3
Pricing Tiers	4
Updates	4
Interface Methods	4
API Methods	5
CalculateP2PRoute	5
GetP2PRouteResponse	5
CalculateTSPRoute	6
GetTSPResponse	6
OptimiseOrders	7
GetOptimisedOrdersResponse	7
OptimiseOrdersAdvanced	8
OptimiseCollectDeliverOrders	8
OptimiseOrdersAdvanced	9
Data Types	10
LatLon	10
Address	10
TimeWindow	10
DepotNode	10
VehicleOption	10
Vehicle	11
EvProfile	11
EvBatteryType	11
EvChargeValues	11
EvConnectors	11
EvConnectorType	12
EvChargeMode	12
EvChargeMethod	12
Service Order	12
CollectDeliverOrder	13

OrderType	13
Options	13
evStrategy	14
evChargeNetwork	14
VehicleType	14
RouteArea	14
RestBreakType	14
CustomBreak	15
Route	15
OptimisedOrder	15
RouteEvent	15
RouteEventType	16

FleetFusion Routing API

System Overview

The routing API provides route and scheduling services optimised for fleets.

The solution provides several API methods:

- **Calculate P2P route** (Point to point route calculation)
- **Calculate TSP route** (Single route optimisation with optimal stop sequence)
- **Optimise Orders** (Optimisation with minimal input data and options)
- **Optimise Orders Advanced** (Advanced Optimisation with maximum options)
- **Optimise Collection-Delivery Orders** (Optimisation with minimal input data and options)
- **Optimise Collection-Delivery Orders Advanced** (Advanced Optimisation with maximum options)

The solution offers standard and advanced methods for the optimisation to simplify integration at the development end.

Basic methods require only the minimum order detail and only a single default vehicle configuration.

Advanced methods require a full vehicle list, order types and vehicle rules which require the developers to have these business objects defined and managed in their platform to make full use of the advanced methods.

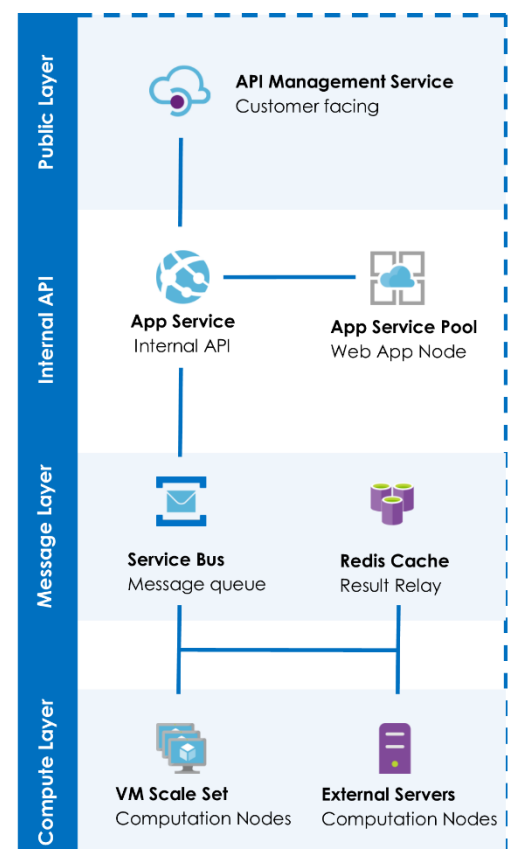
System Architecture

The public API is hosted on Azure, route requests are added to a service bus queue and are then processed by a calculation node.

Each calculation node holds the mapping data sets in RAM for high speed performance calculation. Calculation solutions are then returned to Redis to be picked up by the corresponding get result method.

The calling platform makes the request and is returned Guid for the transaction. The platform is then required to poll the response by calling the get response method. For incomplete request the response will return the progress percentage of the task.

The polling method is required to keep the UI updated with long running tasks, a complete fleet optimisation task with 100 vehicles and 2000 orders can take 3-8 minutes to complete on an 8-core computation node. Average requests complete in less than 30 seconds.



Pricing Tiers

There are two data sources available in the platform which reflect two pricing tiers:

- Enhanced Open Street Map routing data
- HERE premium routing data with historic traffic patterns and optional truck attribute restrictions data.

For most cases, the enhanced Open Street Map data is just as good as the Here premium vector data.

The service cost for the API:

Open Street map data

- Free – limit 10 orders per day
- £5 per month, per vehicle (Optimised solutions are limited to the maximum vehicles)

HERE premium street map data

- £8 per month, per vehicle per month

Here premium street map data with truck attribute restrictions

- £15 per month, per vehicle

These prices are extremely competitive compared to comparative services for example:

- OptimoRoute lowest product is \$17.10 per vehicle per month
- Routific lowest product is \$39 per vehicle per month
- Route4Me lowest product \$299 for fleet routing with only 10 vehicles, additional vehicle \$50 per month

There are other services are various pricing but most do not offer custom vehicle profiles with rules, time window support and multiple vehicle start locations.

Updates

Open Street map data is updated every three months, HERE data updates are issued every three months, but they do not always contain relevant changes to all region networks.

Interface Methods

The API is available as a WEBAPI and SOAP 1.1 and 1.2 methods.

All methods require a customer API key that will be issued upon account creation.

API Methods

CalculateP2PRoute

This generates a simple route between two points.

URI Parameter

APIKey (Required)

Body Parameters

- | | | |
|---------------------|--------------------|--|
| • startPoint | LatLon | lat, lon decimal |
| • endPoint | LatLon | lat, lon decimal |
| • startTime | date | Example 2020-01-29T04:58:12.2900575+00:00 |
| • vehicleType | VehicleType | Car, Van, Truck, Bike – Sets the route profiler used |
| • speedModification | decimal number | 1 is normal, 1.1 is 10 % faster 0.9 is 10% slower |

Returns

A string GUID used to retrieve the response to the request

GetP2PRouteResponse

This method gets the status and result of a submitted request.

URI Parameter

guid (Required)

Returns

P2PResponse

- | | | |
|--------------------|-----------------------------|------------------------------------|
| • complete | Boolean | |
| • error | boolean | |
| • errorDescription | string | |
| • progress | integer | 0-100 for in progress calculations |
| • startPoint | LatLon | |
| • endPoint | LatLon | |
| • startTime | date | |
| • endTime | date | |
| • points | Collection of LatLon | |

CalculateTSPRoute

This takes a route start/end point and a list of stops, it then calculates the optimum route to visit all the stops in the most efficient way.

URI Parameter

APIKey (Required)

Body Parameters

- | | | |
|------------------|-----------------------------|--|
| • startPoint | LatLon | lat, lon decimal |
| • endPoint | LatLon | lat, lon decimal |
| • routeStartTime | date | Example 2020-01-29T04:58:12.2900575+00:00 |
| • vehicleType | VehicleType | Car, Van, Truck, Bike – Sets the route profiler used |
| • stops | Collection of LatLon | |

Returns

A string GUID used to retrieve the response to the request

GetTSPResponse

This method gets the status and result of a submitted request.

URI Parameter

guid (Required)

Returns

TSPResponse

- | | | |
|--------------------|-----------------------------|------------------------------------|
| • complete | boolean | |
| • error | boolean | |
| • errorDescription | string | |
| • progress | integer | 0-100 for in progress calculations |
| • startPoint | LatLon | |
| • endPoint | LatLon | |
| • startTime | date | |
| • endTime | date | |
| • points | Collection of LatLon | |

OptimiseOrders

This method takes the depot as the starting point of all the vehicles, it takes a single vehicle definition, and a list of service orders.

Service orders are an order that only has a single address point, it can be a delivery or a service call.

URI Parameter

APIKey (Required)

Body Parameters

- orders Collection of **ServiceOrder**
- defaultVehicle **Vehicle**
- options **Options**

Returns

A string GUID used to retrieve the response to the request

GetOptimisedOrdersResponse

This method gets the status and result of a submitted request.

URI Parameter

guid (Required)

Returns

OptimisedResponse

- complete boolean
- progress integer
- error boolean
- errorDescription string
- routes Collection of **Route**
- allocatedVehicles Collection of **Vehicle**

OptimiseOrdersAdvanced

This method takes the depot as the starting point of all the vehicles but allows vehicle start and end positions to be set, if the vehicle position is not set, the depot location is used. This method takes a full vehicle list to represent the vehicles of the fleet.

Service orders are an order that only has a single address point, it can be a delivery or a service call.

URI Parameter

APIKey (Required)

Body Parameters

- orders Collection of **ServiceOrder**
- defaultVehicle **Vehicle** (if more routes are generated than vehicles exist, new routes are built with the Default Vehicle specification)
- vehicles Collection of **Vehicle**
- orderTypes Collection of **OrderType** (specifies the order types to match orders to vehicle rules)
- options **Options**

Returns

A string GUID used to retrieve the response to the request

OptimseCollectDeliverOrders

This method takes the depot as the starting point of all the vehicles, it takes a single vehicle definition, and a list of CollectDeliverOrders.

Collect/Deliver orders are an order that has a collection location and a delivery location as part of the same order so both must be completed by the same vehicle and in the correct sequence.

URI Parameter

APIKey (Required)

Body Parameters

- orders Collection of **CollectDeliverOrder**
- defaultVehicle **Vehicle**
- options **Options**

Returns

A string GUID used to retrieve the response to the request

OptimiseOrdersAdvanced

This method takes the depot as the starting point of all the vehicles but allows vehicle start and end positions to be set, if the vehicle position is not set, the depot location is used. This method takes a full vehicle list to represent the vehicles of the fleet.

Collect/Deliver orders are an order that has a collection location and a delivery location as part of the same order so both must be completed by the same vehicle and in the correct sequence.

URI Parameter

APIKey (Required)

Body Parameters

- orders Collection of **CollectDeliverOrder**
- defaultVehicle **Vehicle** (if more routes are generated than vehicles exist, new routes are built with the Default Vehicle specification)
- vehicles Collection of **Vehicle**
- orderTypes Collection of **OrderType** (specifies the order types to match orders to vehicle rules)
- options **Options**

Returns

A string GUID used to retrieve the response to the request

Data Types

LatLon

- lat decimal number
- lon decimal number

Address

- street string
- addressLine2 string
- city string
- region string
- postalCode string
- country string

TimeWindow

- enabled boolean
- startTime date (The earliest time the location can be visited)
- endTime date (The latest time the location can be visited)

DepotNode

- name string
- address **Address**
- point **LatLon**
- addressID integer (This is your ID provided for reference)

VehicleOption

- Option string (Rule label)

Vehicle

- id string
- name string
- options Collection of **VehicleOption**
- weightLimitKg decimal number
- volumeLimitM3 decimal number
- palletLimit decimal number
- relativeSpeed decimal number
- startLocation **DepotNode**
- endLocation **DepotNode**
- **evProfile** **EvProfile**

EvProfile

- maxRange decimal number
- wattHoursPerMile decimal number
- tempSag decimal number
- tempExponent decimal number
- heatingPower decimal number
- coolingPower decimal number
- batteryType **evBatteryType**
- chargeTable collection of **evChargeValues**
- connectors collection of **evConnectors**

EvBatteryType

enumeration

- lithiumIon
- nickelMetalHydride
- leadAcid
- leadCarbon

EvChargeValues

- chargeTimeMins decimal number
- chargePercent decimal number

EvConnectors

- connectorType **evConnectorType**
- voltage integer
- chargeCurrent integer
- chargeMode **evChargeMode**

- chargeMethod **evChargeMethod**

EvConnectorType

enumeration

- TypeG 3-pin Type G (BS1363)
- JEVS JEVS G105 (CHAdeMO) DC
- SAEJ1772 Type 1 SAEJ1772 (IEC 62196)
- Combo Type 2 Combo (IEC62196) DC
- Mennekes Type 2 Mennekes (IEC62196)
- Tesla Type 2 Tesla (IEC62196) DC
- Scame Type 3 Scame (IEC62196)
- Commando3P Commando 3P+N+E (IEC60309)

EvChargeMode

enumeration

- AC
- EVSE
- DC

EvChargeMethod

enumeration

- DC
- SinglePhaseAC
- TrheePahseAC

Service Order

- id string (This is your ID)
- reference string (This is your reference string)
- address **Address**
- point **LatLon**
- workType string (This is the order type label)
- OrderDurationMin decimal number
- timeWindow **TimeWindow**
- packageWeightKg decimal number
- packageVolumeM3 decimal number
- pallets decimal number

CollectDeliverOrder

- id string (This is your ID)
- reference string (This is your reference string)
- collectAddress **Address**
- collectPoint **LatLon**
- collectWorkType string (This is the order type label)
- collectTimeWindow **TimeWindow**
- deliverAddress Address
- deliverPoint **LatLon**
- deliverWorkType string (This is the order type label)
- deliverTimeWindow **TimeWindow**
- deliverImmediately boolean
- packageWeightKg decimal number
- packageVolumeM3 decimal number
- pallets decimal number

OrderType

- vehicleOptions Collection of **VehicleOption** (List of vehicle rules required for this order type)

Options

- routeStartDateTime date
- depot **DepotNode**
- optimiseVehicleStartTime boolean
- maxSingleRouteTimeMins integer
- breakType **RestBreakType**
- customBreak **CustomBreak**
- flexibleTimeWindows boolean
- relativeSpeed decimal number
- routeDirections boolean
- vehicleType **VehicleType**
- routeArea **RouteArea**
- evRouteStrategy **evStrategy**
- evSubscribedNetworks collection of **evChargeNetworks**
- usePAYGChargePoints Boolean

evStrategy

Enumeration

- none 0
- driverBreak 1 charging stops are only scheduled at drivers breaks
- timeWindowWait 2 charging stops are only scheduled when a vehicle is waiting for a delivery time window to open
- opportunistic 3 charging stops are optimised on both driver breaks, time window wait time, and if an available charging site is near to a delivery/collection location. This requires a two pass optimisation run which takes longer to complete the optimisation

evChargeNetwork

- name string
- maxCost decimal

VehicleType

Enumeration

- car 0
- van 1
- truck2
- bike 3
- ev 4

RouteArea

Enumeration

- UK 1
- EU 2
- US 3

RestBreakType

Enumeration

- none 0
- eu_tacho 1
- custom 2

CustomBreak

- durationMins integer
- CanStartFrom date (Earliest time the break can start)
- MustEndBy date (Latest time the break can start)

Route

- vehicleId string
- vehicleName string
- startTime date
- endTime date
- durationMins decimal number
- distance decimal number
- breaksMins decimal number
- orderDurationMins decimal number
- driveTimeMins decimal number
- waitTimeMins decimal number
- orders Collection of **OptimisedOrder**
- events Collection of **RouteEvent**
- points Collection of **LatLon**
- requiredVehicleOptions Collection of **VehicleOptions**

OptimisedOrder

- orderID string
- arriveOnSite date
- workStart date
- workEnd date
- waitDurationMins decimal number
- orderDurationMins decimal number
- timeWindows **TimeWindow**
- withinTimeWindow boolean
- breakDurationMins decimal number
- point **LatLon**

RouteEvent

- description string

- | | |
|--------------------------|-----------------------|
| • eventType | RouteEventType |
| • eventTime | date |
| • orderDurationWorkMins | decimal number |
| • orderDurationTotalMins | decimal number |
| • orderDurationWaitMins | decimal number |
| • orderID | string |
| • breakTimeMins | decimal number |
| • point | LatLon |

RouteEventType

Enumeration

- | | |
|-------------|---|
| • start | 0 |
| • restBreak | 1 |
| • order | 2 |
| • end | 3 |
| • recharge | 4 |