V2

# REST Service: FleetSolver-Retrieve

ROUTEPERFORM.COM

ROUTE PLANNING WEB SERVICES FOR DEVELOPERS

# Welcome

Welcome to the FleetSolver web service reference guide. This solver is our most powerful tool to solve fleet-level routing problems that can involve dozens of routes and hundreds of stops. There are two web services, one to submit and one to retrieve. This document covers the retrieve portion. Both web services are straightforward, powerful, dependable, and allow web developers to rapidly add critically important route planning capabilities to their solution(s).

This document provides information related to this particular web service only. Please keep in mind that we offer a variety of pre-built web services. We may also enhance existing web services or develop entirely new services case-by-case. Contact us for more information.

The documentation is intended as a comprehensive reference manual. As a means to jump-start your implementation we'd also recommend our:

- Interactive SDK
- Code examples – ready to run code for various platforms

# Before You Begin

API key - To gain those lavish accolades from your bosses, customers, peers and groupies, you first need an API key. Contact us for your key to get started.

Geocoding - We require all your input stops to be geocoded (possess lat/lon coordinates). If your addresses are not currently geocoded, you will not be able to pass them to our service.

We are agnostic as to what digital map you prefer to use. As long as you have geocoded data you can tie-in to our service seamlessly.

Geographic Data Coverage – We support all of North America, the UK, most all of mainland Europe, Australia & New Zealand, a good portion of Asia and south America and portions of Africa as well. If you have any questions about your locale please contact us.

Encryption - We require all web traffic be passed as encrypted (HTTPS & TLS1.2).

rest/json - The SingleRouter call is a RESTful web service. We use JSON as input and output.

# Service Overview & Purpose

This document covers the 'Retrieve' of a routing request that was previously sent via the 'Submit' process. You would of course need to read the 'submit' document and implement a submit in order to get as far as to retrieve the results via this web service.

Each submit & retrieve have a routingJobID in common. The jobs themselves may take several seconds or several minutes to complete.

The 'Retrieve' call both informs you if the job completed, and also returns results, once completed. You may wish to call the 'Retrieve' on a timed process to continue to check if the job has completed. The polling to determine the job status should be called no more than every 10 seconds. Any calls made more frequently will result in the cancellation of the job.

Results are kept for 24 hours once finalized. You can retrieve them several times within that 24-hour period although typically only 1 retieve is required, of course.

Jobs that cannot complete return with an error code to help you to diagnose the issue at hand.

It is very possible that some (or all) stops aren't routed. In such cases you can inspect them in the unrouted area along with the violated constraints that prevented the route inclusion for each stop item.

# Requests

This service accepts POST requests via HTTPS. The parameters and values are transferred in the body of the request as JSON. This service runs asynchronously. The 'Submit' call will send the job to the service and the 'Retrieve' call will later be called to gather results. Syntax required is outlined below. Jobs can take up to several minutes to complete. We ask that you check on the retrieve request no more than every 30 seconds.

### Sample Code: Request URI

https://www.routeperform.com/services/v1/fleet-solver-submit/ (noted here, but documented elsewhere)

https://www.routeperform.com/services/v1/fleet-solver-retrieve/

see below for information on body parameters

# Retrieve Request Parameters

The body text of the request will contain all input parameters. This body text needs to be JSON-formatted.

About JSON:
https://en.wikipedia.org/wiki/JSON

### Sample Request: An Example

```
{
    "apiKey": "{yourKey}",
    "passthroughGUID": "{GUID}",
    "routingJobID": "{IDReturnedfromSubmitServiceResponse}",
}
```

## Retrieve Request - General Parameters

| Parameter name | Type | Description |
|---|---|---|
| apiKey | string | Your unique authentication token gathered from our portal. |
| passthroughGUID | string | A GUID provided to uniquely identify each request. You may also pass this as a request header (recommended). The GUID should be truly unique, do not re-use the GUID from the submit request. |
| routingJobID | string | The 'submit' web service's response will provide you with this ID that then is supplied as input for your 'retrieve' request. |

# Retrieve Response – Overview

Review the HttpStatusCode first. Value 200 'OK' confirms the service returned a response but does not verify that the route activity could be returned. In the event such as bad input data it could for instance return a 200 'OK' but lack route results.

The possible HttpStatusCode return values are plentiful. For example: https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

The body text of the response will contain all output results. This body text returned will be JSON-formatted.

The HTTP Content-Type is "application/json;charset=UTF-8"

about JSON:
https://en.wikipedia.org/wiki/JSON

The resultCode value will verify the success or failure of the request. Appendix B lists all possible result codes.

## Sample Response


```
{
    "passThroughGUID": "{GUIDfromRequest}",
    "outcome": {
      "resultCode": 2000,
      "resultCodeDesc": "Successfully retrieved",
      "routingJobID": "{fromRequest}",
      "routingJobStatus": "JobSucceeded"
    },
    "outputRoutes": [
       {
          "routeID": "123",
          "routeDisplayName": "Joe Smith",
          "totalMeters": 51981.81,
          "totalMiles": 32.3,
          "totalMinutes": 172.3,
          …
          "routedItems": [
              {
                  "stopType": 1,
                  "stopID": "1",
                  "stopDisplayName": "HQ",
                  "latitudeY": 32.728328,
                  "longitudeX": -117.171133,
                  "itemSequence": 1,
                  "stopSequence": "1",
                  "serviceMinutes": 5,
                  …
              },…
          ],
          "polyline": [
              {
                  "latY": 32.48828,
                  "lonX": -117.48828
              },…
          ]
       },…
    ],
    "unroutedItems": [
          {
              "stopType": 3,
              "stopID": "95753",
              "stopDisplayName": "Main Street Station",
              "latitudeY": 32.223328,
              "longitudeX": -117.564133,
              "violatedConstraints": "B,N",
              "violatedConstraintsText": "reason1|reason2"
          },…
    ]
}
```

# Retrieve Response - Content

The body text of the response will contain all output results. This body text returned will be JSON-formatted.

Response - General output

| Element name | Type | Description |
|---|---|---|
| passthroughGUID | string | A GUID provided to uniquely identify each request that is output in the response. |
| outputRoutes | (see below) | |
| unruotedItems | (see below) | |

## RESPONSE-OUTCOME

| Element name | Type | Description |
|---|---|---|
| resultCode | integer | See Appendix B for full information. The result code is specific to if the 'retrieve' service could complete as expected. Consult the routingJobStatus for information pertaining to the success of the internal routing job itself. If the status indicates the processing is ongoing them you can wait 10+ seconds and then perform the request again until the status finally indicates success or failure. |
| resultCodeDesc | string | Text that is a readable representation of the result code returned. |
| routingJobId | string | The 'submit' web service's response will provide you with this ID that then is supplied as input for your 'retrieve' request and subsequently passed-back within the response. |
| routingJobStatus | string | See Appendix C for full information. You should only utilize this value if the resultCode returned success. |
| routesCount | integer | Only routes that were assigned 1 or more route stops will be returned and included within this count. |
| routedStopsCount | integer | Stops only, not inclusive of depots, breaks, etc. |
| unroutedStopsCount | Integer | Stops only, not inclusive of depots, breaks, etc. |

## RESPONSE-OUTPUTROUTES

| Element name | Type | Description |
|---|---|---|
| routeID | string (50) | |
| routeDisplayName | string (50) | |
| totalMeters | double | |
| totalMiles | double | |
| totalServiceMinutes | double | |
| totalDriveMinutes | double | |
| totalEarlyMinutes | double | Total minutes of early arrival accrued waiting for time windows to become available. Early minutes are unproductive but are a necessity at times as they are less costly than departing and returning. |
| totalBreakMinutes | double | |
| totalRefreshMinutes | double | Minutes, if any, spent refreshing the vehicle in terms of capacities filling or emptying. |
| totalMinutes | double | Service minutes + drive minutes + break minutes + early minutes + refresh minutes. Late minutes aren't included as they are noteworthy but don't actually accrue time. |
| totalLateMinutes | double | A note of the total minutes of lateness for time windows. |
| totalItemsCount | integer | Count of stops, breaks and depots in total on this route. |
| routedStopsCount | integer | Stops only for this route, not inclusive of depots, breaks, etc. |
| totalCapacityToDeliver1 | double | |
| totalCapacityToPickUp1 | double | |
| polyline | (See below) | |

## RESPONSE-ROUTEDITEMS

| Element name | Type | Description |
|---|---|---|
| stopID | string | |
| stopDisplayName | string | |
| stopType | integer | Valid values:<br>• 0 – None<br>• 1 – Start Depot<br>• 2 – End Depot<br>• 3 – Stop<br>• 4 – Break<br>• 5 – Refresh |
| latitudeY | double | Latitude potion of the geographic coordinate that was supplied with the request. Example: 32.708328 |
| longitudeX | double | Longitude potion of the geographic coordinate that was supplied with the request. Example: -117.161133 |
| itemSequence | integer | Present for all items including stops, depots, breaks, etc. |
| stopSequence | string | Only present for stopType = 3. This provides the sequenced value of the routed stops and only the routed stops. |
| serviceMinutes | double | The pass-through service minutes for a routed stop, or the break minutes for a break. |
| earlyArrivalMinutes | double | Minutes (if any) of early arrival awaiting a time window to become available. |
| lateArrivalMinutes | double | Minutes (if any) of late arrival in relation to the time window. |
| driveMinutesFromPrevious | double | |
| metersFromPrevious | double | |
| etaUTC | string | Estimated time of arrival in UTC time in yyyymmdd hhmmss. |
| etdUTC | string | Estimated time of departure in UTC time in yyyymmdd hhmmss. |
| etaEpochUTC | long | Estimated time of arrival in UTC time in epoch format. See Appendix A for time format information. |
| etdEpochUTC | long | Estimated time of departure in UTC time in epoch format. See Appendix A for time format information. |
| CapacityToDelivery | double | |
| capacityToPickUp | double | |

## RESPONSE-POLYLINE

Note: The polyline is an array of coordinates that create a visual representation of each route path traveled.

| Element name | Type | Description |
|---|---|---|
| latY | double | Latitude potion of the geographic coordinate that will, in totality, create a polyline to show the route shape. Example: 32.708328 |
| lonX | double | Longitude potion of the geographic coordinate that will, in totality, create a polyline to show the route shape. Example: -117.161133 |

## RESPONSE – UNROUTEDITEMS

| Element name | Type | Description |
|---|---|---|
| stopType | string | Valid values:<br>• 0 – None<br>• 1 – Start Depot<br>• 2 – End Depot<br>• 3 – Stop<br>• 4 – Break<br>• 5 - Refresh |
| stopID | | |
| stopDisplayName | | |
| latitudeY | double | Latitude potion of the geographic coordinate that was supplied with the request. Example: 32.708328 |
| longitudeX | double | Longitude potion of the geographic coordinate that was supplied with the request. Example: -117.161133 |
| violatedConstraints | string | (see appendix D) Comma delimited as there may be 1 or more responsible constraints. |
| violatedConstraintsText | string | (see appendix D) Pipe delimited as there may be 1 or more responsible constraints. |

# Appendix A – Output Times

Handling times is tricky, mainly because route plans can span time zones. Route plans can also span days (past midnight and beyond). Time of day also matters for traffic-related aspects. The simple system described below must be used for formatting times to safeguard against any ambiguity related to dates and times.

Please review the 'Submit' document for full information on the formatting and examples of specific input times. The input times will influence the output times, of course. Generally speaking, all output times are provided in UTC. In most cases the output times are provided both in epoch (machine-readable) and traditional (human-readable) formats.

For converting values to/from epoch, you may wish to utilize this website as a resource: https://www.epochconverter.com/

# Appendix B – Result Codes

resultCodes

| Result Code | Value | Notes |
|---|---|---|
| SuccessfullyRetrieved | 2000 | Not to be confused with the http result code of 200. |
| NotProcessed | 0 | |
| ErrorNoAPIKeySupplied | 1 | |
| ErrorInvalidAPIKeySupplied | 2 | |
| ErrorNoRoutingJobIDSupplied | 3 | |
| ErrorInvalidRoutingJobIDSupplied | 4 | |
| ErrorInvalidRequestSupplied | 10 | Please verify your JSON data is in valid format. |
| ErrorDuringPreValidation | 20 | General error when validating input data. |
| ErrorDuringRoutingJobStatusDetermination | 25 | Internal |
| ErrorReturnedForRoutingJob | 26 | Internal |
| ErrorDuringRoutingJobStatusDeterminationHandling | 27 | Internal |
| ErrorDuringProcessingJobStatusCheck | 28 | Internal |
| ErrorDuringProcessingRetrievingResultStops | 29 | Internal |
| ErrorReturnedForRoutingJobFetchStopsPortion | 30 | Internal |
| ErrorDuringFetchingStopsHandling | 31 | Internal |
| ErrorDuringFetchStopsProcedure | 32 | Internal |
| ErrorDuringSortingReturnStops | 33 | Internal |
| ErrorDuringInternalProcessingWhileRetrievingResultStops | 34 | Internal |
| ErrorDuringProcessingRetrievingRouteSummaryInfo | 40 | Internal |
| ErrorDuringInternalProcessingWhileRetrievingResultRoutesSummaryInfo | 41 | Internal |
| ErrorReturnedForRoutingJobFetchRouteSummaryPortion | 42 | Internal |
| ErrorDuringFetchingRouteSummaryHandling | 43 | Internal |

| | | |
|---|---|---|
| ErrorDuringFetchRouteSummaryProcedure | 44 | Internal |
| ErrorDuringFetchRouteGeometryProcedure | 54 | Internal |
| ErrorDuringProcessingWhileRetrievingUnrouted | 60 | Internal |
| ErrorReturnedForRoutingJobFetchUnroutedPortion | 61 | Internal |
| ErrorDuringFetchingUnroutedStopsHandling | 62 | Internal |
| ErrorDuringFetchUnroutedStopsProcedure | 63 | Internal |

# Appendix C – Routing Job Status Codes

routingJobStatus Codes

| Result Text | Notes |
|---|---|
| failedToProcess | Check output messages and unrouted stops constraint violations for additional information. |
| stillProcessing | Please wait at least 10 seconds before trying again. |
| completedSuccessfully | |

# Appendix D – Violated Constraint Codes

violatedConstraints

| Result Code | Readable Constraint |
|---|---|
| A | Unknown constraint. |
| B | Break maximum work time violation. |
| C | Break maximum travel time violation. |
| D | Refresh could not be inserted along target route without introducing constraint violations. |
| E | Break could not be inserted along target route without introducing constraint violations. |
| F | Stop is unreachable from previous stop. |
| G | Stop cannot be routed due to constraint violation on paired stop. |
| H | Stop pair constraint violation. |
| I | Stop does not fall within the hard route zone of the target route(s). |
| J | Rules required by an order were not found on any target route(s). |
| K | Stop, depot, refresh, or break hard time window constraint violation. |
| L | Route MaxTotalDistance constraint violation. |
| M | Route MaxTotalTravelTime constraint violation. |
| N | Route MaxTotalTime constraint violation. |
| O | Route Capacities constraint violation. |
| P | Route MaxOrderCount constraint violation. |