RouteSavvy API PLUS Documentation

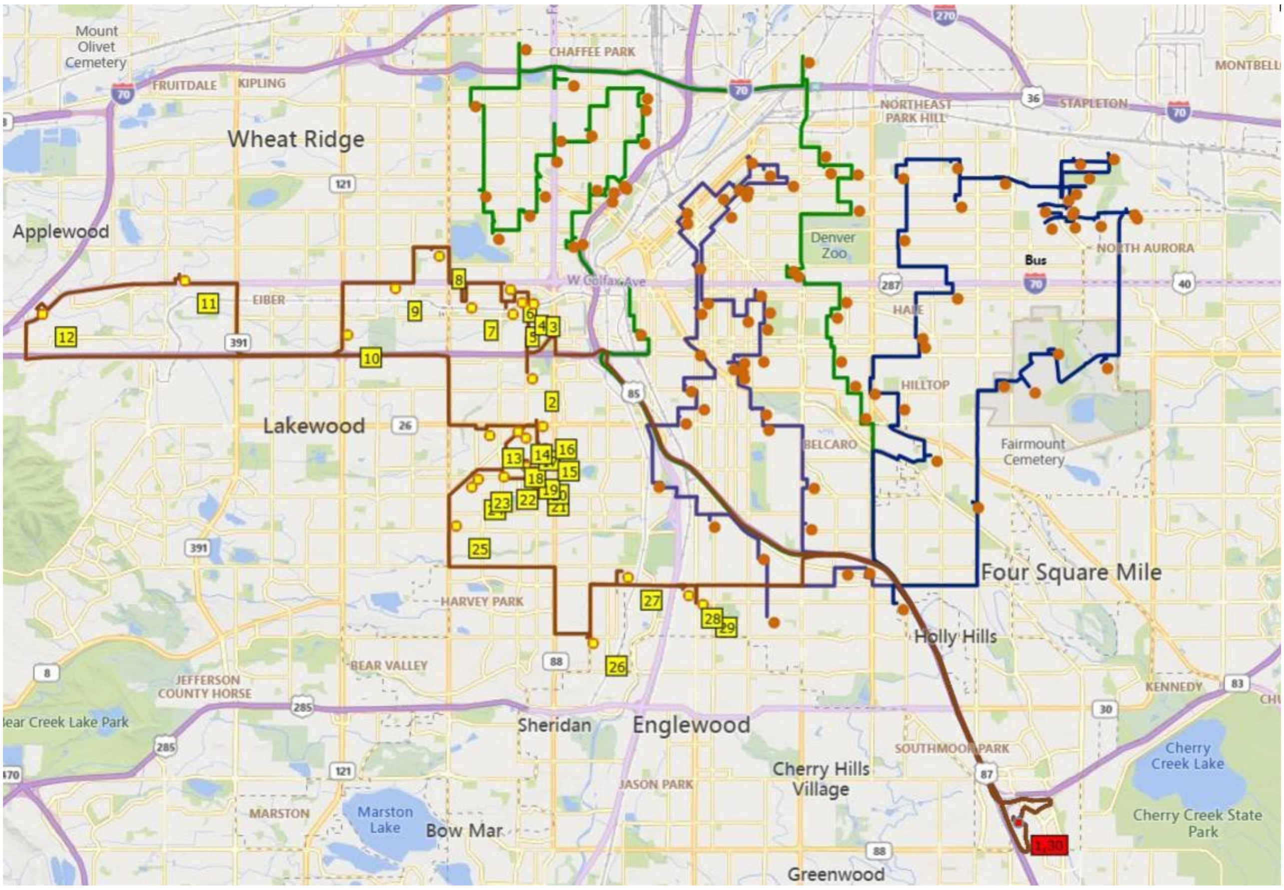
Introduction:

RouteSavvy API PLUS is an easy-to-use REST service for route planning and optimizing driving stops across multiple vehicles and multiple route stop locations. In addition to calculating optimized routes based on stop locations and vehicle start and end locations, RouteSavvy API PLUS supports time window scheduling, driver availability, driver breaks, and stop location priority levels. The optimized results return multiple routes each with stops ordered in the most efficient driving route according to driver availability. Results include turn-by-turn directions for each route leg as well as latitude and longitude polylines so the routes can be used in map visualizations. Multiple vehicles that are available for driving their own unique route may be defined. A single vehicle may be defined as input, however creating more than one route within the same optimization request is the primary purpose of this API.

RouteSavvy API PLUS supports GET and POST request options. Requests use JSON formatted strings and results are returned in JSON format for optimal use in web applications. The API supports a maximum of 1,000 locations (route stops) optimized across a maximum of 50 vehicles (resultant routes). Trial/dev/test access is limited to a maximum of 100 locations and 5 vehicles, with limitations on daily and aggregate optimization requests.

Production access is offered through varying license tiers based on maximum numbers of locations/vehicles, associated costs are usage-based and transactionally dependent on the numbers of both locations and vehicles per optimization request. Please contact info@routesavvy.com for more information on licensing and pricing.

Here's an example of optimized multi vehicle routes from RouteSavvy API PLUS used in a map visualization.



API Endpoint:

GET Url: https://api.routesavvy.com/plus/RSAPI.svc/GETOptimize?query={jsonquery}
POST Url: https://api.routesavvy.com/plus/RSAPI.svc/POSTOptimize

Note:
GET requests (RSAPI.svc/GetOptimize) requests are limited to the URL encoded parameter maximum length. Although this varies by browser, 2,048 characters is the typical suggested limit for encoding stops in the URL string. Because of the URL length maximum, you will be limited to a smaller number of customer stops that can be optimized using this approach. POST requests (RSAPI.svc/PostOptimize) requests do not have URL length limitations and can accommodate larger numbers of customer stops.

API Request Input Parameters:

There are two primary components of the API request, *Locations* array and *OptimizeParameters*. The *Locations* array are the input details about the customer stop locations and the *OptimizeParameters* are the input details about the vehicles that will visit the customer stop locations.

***Locations*:**

- **Name:**
  This is the custom name you can specify for the customer stop locations. If no Name is defined for a stop location, results will identify "Name" as *null* and the location can only be identified by its Latitude/Longitude.

- **Latitude/Longitude:**
  Latitude/longitude for the specified customer stop locations and the start and end locations for the definition of vehicle/driver parameters. 6-decimal latitude/longitude accuracy is recommended to get the best optimization results. Lower accuracy latitude/longitude data may be used but may cause an error in API processing as well as inaccurate or unusable route results. If you do not have latitude/longitude data and only have street addresses, any address geocoding service can be used to convert street address data to latitude/longitude.
  OnTerra Systems can provide access to commercial geocoding services if needed. Please contact info@routesavvy.com for licensing and pricing information related to available geocoding services.

- **VisitDurationInMinutes:**
  The API also incorporates expected stop duration or the amount of time expected to complete each stop, defined in the *VisitDurationInMinutes* parameter. Visit duration specifies how long a visit, such as delivery, pick-up or repair may take at the customer stop location. If no stop duration is defined, then default of 0 minutes will be assumed. *VisitDurationInMinutes* is optional but is recommended if scheduling is critical to optimization needs. Please note, a stop may be assigned to a specific time window as long as the resultant beginning time of the stop falls within that time window. It is possible that the resultant ending time of the stop will occur after the close of the time window. The optimization only considers that the stop begins within the assigned time frame, and the stop duration may run after the close of the time window. For vehicle start and end locations, *VisitDurationInMinutes* is not recommended since it typically doesn't apply to such locations.

- **Time_Window:**
  Time_Window specifies the time-of-day that each route stop may be completed from the earliest to the latest time. Time windows represent time constraints at a stop such as the earliest and latest a customer stop location is open for deliveries or is staffed. Multiple time windows may be defined for

any stop. Stop time windows are defined in HH:MM format and limited to a 24-hour period (00:00 to 23:59). If no stop time parameters are passed, then the stop time window will be defined as 24 hours (00:00 to 23:59). For vehicle start and end locations, *Time_Window* is not recommended since it typically doesn't apply to such locations.

- **Priority:**
  The *Priority* parameter allows you to specify the importance of each stop location using a scale from 1 (highest priority) through 10 (lowest priority). *Priority* is only used when the optimization service cannot assign all stops to a route due to tight constraints related to Driver Availability, Time Windows, or Visit Duration. In this situation, higher priority stops will be selected for the route, and lower priority stops may be skipped. If all stops can be included in the route while satisfying all the constraints, then the *Priority* parameter doesn't influence the optimization results. The route optimization process always satisfies hard constraints (i.e.: Driver Availability, Time Windows and Visit Duration) first before using *Priority* to determine which stops to keep or skip. *Priority* is an optional parameter. If a priority isn't specified for a given stop location, then a default priority value of 2 is assigned to the stop location.

***OptimizeParameters*:**

- **AppID:**
  AppID is your credentials (alphanumeric token) for accessing the API. If you are a current user of our legacy RouteSavvy API, your existing AppID can be used for trial/dev/test access to RouteSavvy API PLUS. If you do not have credentials, you can register for a free trial and receive credentials any time at the following link: https://www.routesavvy.com/routesavvy-api-free-trial/
- **OptimizeType:**
  OptimizeType specifies whether "time" or "distance" should be used to calculate the travel optimization. RouteSavvy API PLUS performs optimization based on the actual road network between locations, including both travel distance and travel time (based on varying travel speeds of various road types), or what we refer to as "RealRoad" optimization. The "time" option will produce the optimization result with the shortest drive time, while "distance" will produce the result based purely on the shortest driving distance.
  The routing used is based on standard vehicle driving directions. If more advanced routing parameters are required, such as the ability to avoid highways, avoid tolls, avoid certain commercial vehicle or truck routing restrictions, such as hazmat or vehicle dimension (height, width, weight) restrictions, OnTerra Systems also offers this through our RouteSavvy API PLUS PREMIUM. Please contact info@routesavvy.com for more details including information on licensing and pricing for advanced routing capabilities.

- **MinimizeVehicles:**
  The *MinimizeVehicle* parameter allows for the explicit creation of only the minimal number of routes required to complete the stops passed for optimization. This disregards the number of vehicles specified in the API request. For example, if 5 vehicles are input, and all of the stops can be reached using 3 vehicles and their assigned availability, then the result will include routes for 3 vehicles. *MinimizeVehicle* is a boolean option (true or false). If the *MinimizeVehicle* parameter is not used (i.e.: =false), then the stops passed for optimization will be distributed across all defined vehicles. This may result in routes not utilizing the full driver availability time of some or all vehicles/drivers. If more vehicles are included in the input than are required to complete all of the route stops (locations), then using *MinimizeVehicle* (i.e.: =true) will return a result that accomplishes the routing with the minimal number of vehicles.

- **Vehicles:**
  Vehicles are defined by the following components: *vehicle_id*, *StartLocation*, *EndLocation*, *DriverAvailabilityFrom*, *DriverAvailabilityTo* and *Break*

  All vehicle *StartLocation* and *EndLocation* must be defined in the Locations array. All vehicles may be defined with the same start and end locations (for example all vehicles start and end at the same depot or warehouse), or each vehicle may be defined with its own unique start and end locations. Note, if a location in the Locations array is used as a vehicle's *StartLocation* or *EndLocation* then that particular location is designated for that specific purpose and is not factored into the route optimization as a stop for any other vehicles in the request (unless other vehicles also use the location as a start/end).
  The time-of-day availability of each vehicle/driver should be defined using the *DriverAvailabilityFrom* and *DriverAvailabilityTo* parameters. Availability time is defined in HH:MM format and limited to a 24-hour period (00:00 to 23:59). If *DriverAvailabilityFrom* and *DriverAvailabilityTo* are not passed, then the vehicle/driver availability will be defined as 168 hours by default (i.e. 7 days or one week, which can be used for long-haul routing scenarios).

  The driver *Break* parameter allows for specifying a driver break to be scheduled between a driver's *DriverAvailabilityFrom* and *DriverAvailabilityTo*. The *Break* parameter consists of the following details:

    - *Earliest*: Earliest time the driver can take a break
    - *Latest*: Latest time the driver can take a break
    - *Duration*: Duration of the break (in minutes)

  The driver break time is defined in HH:MM format and limited to a 24-hour period (00:00 to 23:59). The driver break is factored into the route optimization calculation as an operational constraint similar to the *DriverAvailabilityFrom* and *DriverAvailabilityTo* parameters. A driver break influences the route optimization stop order, timing and whether stops can be achieved. If all the driver's stops are completed before the specified driver break, then the break is skipped. *Break* is an optional parameter. Only one break can be specified per driver (vehicle) in the API request.

API Request Input Parameter Details:

Parameter Name	**Data Type**	**Required?**	**Description**	**Example**
Locations array: *Name*	string	No	Custom name value for the customer stop location.	"Customer01"
Locations array: *Latitude*	float	Yes	Latitude value for the stop location and the vehicle start and end locations.	"39.595140"
Locations array: *Longitude*	float	Yes	Longitude value for the stop location and the vehicle start and end locations.	"-104.849620"
Locations array: *VisitDurationInMinutes*	integer	No	Expected visit duration for the stop location, in minutes.	"10"
Locations array: *Time_Window: earliest*	string	No	The earliest time a stop can be visited.	"09:00"
Locations array: *Time_Window: latest*	string	No	The latest time a stop can be visited.	"12:00"
Locations array: *Priority*	integer	No	A value from 1 (highest priority) to 10 (lowest priority) assigned to a stop location to indicate importance when tight vehicle or stop location constraints may require some stop locations to be skipped in the optimization calculation. If not specified, "2" is the default.	"1"
OptimizeParameters: *AppID*	String	Yes	Your Onterra AppID value to authenticate the API request.	"90c13....21a609"
OptimizeParameters: *OptimizeType*	string	No	Specify whether the route optimization should be based on time or distance. Valid input values are "Time" or "Distance". If not specified, "Time" is the default.	"Time"
OptimizeParameters: *MinimizeVehicle*	boolean	Yes	Boolean "true" or "false".	"true"

OptimizeParameters: Vehicles - *vehicle_id*	string	Yes	Custom name value for the vehicle.	“Vehicle01”
OptimizeParameters: Vehicles - *StartLocation*	string	Yes	The start location of the vehicle.	“Depot”
OptimizeParameters: Vehicles - *EndLocation*	string	Yes	The end location of the vehicle.	“Depot”
OptimizeParameters: Vehicles - *DriverAvailabilityFrom*	string	No	The start time that the vehicle/driver is available from. If not defined, default is a 168-hour window.	“09:00”
OptimizeParameters: Vehicles - *DriverAvailabilityTo*	string	No	The end time that the vehicle/driver is available until. If not defined, default is a 168-hour window.	“17:00”
OptimizeParameters: Vehicles - *Break: Earliest*	string	No, but it must be specified if using the *Break* feature.	Earliest time the driver can take a break. Time is defined in HH:MM format and limited to a 24-hour period (00:00 to 23:59).	“12:00”
OptimizeParameters: Vehicles - *Break: Latest*	string	No, but it must be specified if using the *Break* feature.	Latest time the driver can take a break. Time is defined in HH:MM format and limited to a 24-hour period (00:00 to 23:59).	“13:00”
OptimizeParameters: Vehicles - *Break: Duration*	integer	No, but it must be specified if using the *Break* feature.	Duration of the break (in minutes)	“60”

API Response Output Fields:

The JSON response from the API request includes the route stop details for the optimized route that was generated for each vehicle. Below is a table that has output field details, descriptions and examples returned in the API response.

API Response Output Field Details:

Field Name	Data Type	Description	Example
Message	string	Status of the API response, such as "Success" if successful or an error detailed why the request failed.	"Success"
Route_Details		Includes the following: • Routes	
Routes		Includes the following: • Vehicle_id • OptimizedStops • Route	
Vehicle_id	string	Name of vehicle the route is for.	"Vehicle01"
OptimizedStops		Includes the following info for the Vehicle_id: • Arrival • Departure • Distance • Duration • IsDuplicate • Name • RouteLocation • StopTimeMinutes • time_windows • waiting_time	
OptimizedStops: *Arrival*	string	Date and time of arrival	"17:05:49 5/15/2023"
OptimizedStops: *Departure*	string	Date and time of departure	"17:10:49 5/15/2023"
OptimizedStops: *Distance*	string	Drive distance in miles and kilometers	"3.30 miles (5.31 km)"
OptimizedStops: *Duration*	string	Drive time in minutes and seconds	"5 minutes, 49 seconds"
OptimizedStops: *IsDuplicate*	boolean	Indicates if a specific stop location has been used more than once in a route.	false
OptimizedStops: *Name*	string	Name value for the stop location.	"Customer008"
OptimizedStops: *RouteLocation*	float	Latitude and longitude of the stop.	"Latitude": 39.59503 "Longitude": -104.80432
OptimizedStops: *StopTimeMinutes*	integer	Duration of the stop in minutes.	5
OptimizedStops: *time_windows*	string	Earliest time (HH:MM) and latest time (HH:MM) for the stop.	"earliest": 09:00, "latest": 12:00

OptimizedStops: *waiting_time*	integer	Waiting time between stops, in seconds.	2915
Route		Includes the following: • DrivingDistance • DriveDistanceUnit • DriveTime • DriveTimeUnit • RouteLegs • RoutePath	
Route: *DriveDistance*	float	Total driving distance of the route, in kilometers.	53.088826
Route: *DriveDistanceUnit*	string	Unit of measurement of the driving distance.	“Kilometer”
Route: *DriveTime*	integer	Total driving time of the route, in seconds.	3133
Route: *DriveTimeUnit*	string	Unit of measurement of the driving time.	“Second”
Route: *RouteLegs*		Includes the following: • Directions • DriveDistance • DriveTime • LegBegin • LegEnd	
Route: *RoutePath*	Array of floats	An array of latitudes and longitudes that make up the polyline of route path. This can be used to display the route path on a map.	[39.595137, -104.84962], [39.595134, -104.850088], [39.59559, -104.85009]
RouteLegs: *Directions*	Array of strings	Step by step driving instructions for the route leg.	“Make a U-turn onto South Holly Street”, “Turn left onto East County Line Road”, “Turn right onto South Quebec Street”
RouteLegs: *DriveDistance*	float	Driving distance of the route leg, in kilometers.	2.72656
RouteLegs: *DriveTime*	integer	Driving duration of the route leg, in seconds.	302

RouteLegs: *LegBegin*	Array of a string and a float	Name value of stop location that the leg begins at and the RouteLocation (latitude/longitude) of that stop location.	“Name”: “Customer101“, “RouteLocation”: { “Latitude”: 39.567508, “Longitude”: -104.923069}
RouteLegs: *LegEnd*	Array of a string and a float	Name value of stop location that the leg ends at and the RouteLocation (latitude/longitude) of that stop location.	“Name”: “Customer102”, “RouteLocation”: { “Latitude”: 39.558012, “Longitude”: -104.90667}
UnassignedStops	Array of strings	List of the stops that couldn’t be assigned due to time constraints and vehicle availability, if applicable. It will include detailed info such as name, code, ID and reason.	"Name": "Customer001", "code": "2", "id": "location-1", "reason": "cannot be visited within time window"

Unassigned Stops:
In the event that the availability of defined vehicles/drivers is insufficient to complete all stops passed for optimization, then unassigned stops will be identified and included in the result.

Technical Support:
For all support inquiries, please email support@routesavvy.com and be sure to mention that you are using the RouteSavvy API PLUS and include as many details as possible regarding your inquiry. We’re happy to assist you with any questions or issues you may have.

Example POST API request:

The following POSTOptimize example includes 10 route stops with varying stop durations and time windows, 3 vehicles each starting and ending at the same location with the same time availability, attempts to minimize vehicles, but the stops cannot be completed within the driver availability so some stops are unassigned.

https://api.routesavvy.com/plus/RSAPI.svc/POSTOptimize

```
{
    "Locations": [
        {
            "Name": "Depot",
            "Latitude": 39.712797,
            "Longitude": -104.987282
        },
        {
            "Name": "Customer001",
            "Latitude": 39.761975,
            "Longitude": -104.907292,
            "VisitDurationInMinutes": 90,
            "Priority": 1,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                },
                {
                    "earliest": "15:00:00",
                    "latest": "18:00:00"
                }
            ]
        },
        {
            "Name": "Customer002",
            "Latitude": 39.765176,
            "Longitude": -104.919965,
            "VisitDurationInMinutes": 90,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                },
                {
                    "earliest": "15:00:00",
                    "latest": "18:00:00"
                }
            ]
        },
```

```
        {
            "Name": "Customer003",
            "Latitude": 39.781996,
            "Longitude": -105.021899,
            "VisitDurationInMinutes": 90,
            "Priority": 10,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                },
                {
                    "earliest": "15:00:00",
                    "latest": "18:00:00"
                }
            ]
        },
        {
            "Name": "Customer004",
            "Latitude": 39.750206,
            "Longitude": -104.933919,
            "VisitDurationInMinutes": 90,
            "Priority": 5,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                },
                {
                    "earliest": "15:00:00",
                    "latest": "18:00:00"
                }
            ]
        },
        {
            "Name": "Customer005",
            "Latitude": 39.762898,
            "Longitude": -104.934401,
            "VisitDurationInMinutes": 90,
            "Priority": 5,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                },
                {
                    "earliest": "15:00:00",
                    "latest": "18:00:00"
```

```
                }
            ]
        },
        {
            "Name": "Customer006",
            "Latitude": 39.763766,
            "Longitude": -104.946255,
            "VisitDurationInMinutes": 90,
            "Priority": 1,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                }
            ]
        },
        {
            "Name": "Customer007",
            "Latitude": 39.764119,
            "Longitude": -104.953665,
            "VisitDurationInMinutes": 60,
            "Priority": 10,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                }
            ]
        },
        {
            "Name": "Customer008",
            "Latitude": 39.770528,
            "Longitude": -105.025323,
            "VisitDurationInMinutes": 90,
            "Priority": 1,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                }
            ]
        },
        {
            "Name": "Customer009",
            "Latitude": 39.758191,
            "Longitude": -105.011556,
            "VisitDurationInMinutes": 60,
            "Priority": 10,
```

```
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                }
            ]
        },
        {
            "Name": "Customer010",
            "Latitude": 39.758643,
            "Longitude": -104.981753,
            "VisitDurationInMinutes": 90,
            "Priority": 5,
            "Time_Window": [
                {
                    "earliest": "09:00:00",
                    "latest": "12:00:00"
                }
            ]
        }
    ],
    "OptimizeParameters": {
        "AppId": "Your_API_Key",
        "OptimizeType": "time",
        "MinimizeVehicle": true,
        "Vehicles": [
            {
                "vehicle_id": "Vehicle01",
                "StartLocation": "Depot",
                "EndLocation": "Depot",
                "DriverAvailabilityFrom": "08:00:00",
                "DriverAvailabilityTo": "19:00:00",
                "Break": {
                "Earliest": "11:00",
                "Latest": "12:00",
                "Duration": 30
                        }
            },
            {
                "vehicle_id": "Vehicle02",
                "StartLocation": "Depot",
                "EndLocation": "Depot",
                "DriverAvailabilityFrom": "08:00:00",
                "DriverAvailabilityTo": "19:00:00",
                "Break": {
                "Earliest": "11:00",
                "Latest": "12:00",
                "Duration": 30
```

```
                        }
            },
            {
                "vehicle_id": "Vehicle03",
                "StartLocation": "Depot",
                "EndLocation": "Depot",
                "DriverAvailabilityFrom": "08:00:00",
                "DriverAvailabilityTo": "19:00:00",
                "Break": {
                "Earliest": "11:00",
                "Latest": "12:00",
                "Duration": 30
                        }
            }
        ]
    }
}
```

Example POST API request:

The following shows the RouteSavvy API PLUS partial result from the above example POST request.

```
    "Message": "Success",
    "Route_Details": [
        {
            "Routes": [
                {
                    "Vehicle_id": "Vehicle01",
                    "OptimizedStops": [
                        {
                            "Arrival": null,
                            "Departure": "8:00:00 12/16/2025",
                            "Distance": null,
                            "Duration": null,
                            "IsDuplicate": false,
                            "Name": "Depot",
                            "RouteLocation": {
                                "Latitude": 39.712797,
                                "Longitude": -104.987282
                            },
                            "StopTimeMinutes": 0,
                            "time_windows": null,
                            "waiting_time": 0
                        },
                        {
```

```
            "Arrival": "8:59:59 12/16/2025",
            "Departure": "9:59:59 12/16/2025",
            "Distance": "4.63 miles (7.45 km)",
            "Duration": "8 minutes, 16 seconds",
            "IsDuplicate": false,
            "Name": "Customer009",
            "RouteLocation": {
                "Latitude": 39.758191,
                "Longitude": -105.011556
            },
            "StopTimeMinutes": 60,
            "time_windows": [
                {
                    "earliest": "09:00",
                    "latest": "12:00"
                }
            ],
            "waiting_time": 3103
        },
        {
            "Arrival": "10:11:14 12/16/2025",
            "Departure": "11:11:14 12/16/2025",
            "Distance": "5.56 miles (8.94 km)",
            "Duration": "11 minutes, 14 seconds",
            "IsDuplicate": false,
            "Name": "Customer007",
            "RouteLocation": {
                "Latitude": 39.764119,
                "Longitude": -104.953665
            },
            "StopTimeMinutes": 60,
            "time_windows": [
                {
                    "earliest": "09:00",
                    "latest": "12:00"
                }
            ],
            "waiting_time": 0
        },
        {
            "Arrival": "11:11:14 12/16/2025",
            "Departure": "11:41:14 12/16/2025",
            "Distance": "0.00 miles (0.00 km)",
            "Duration": "0 seconds",
            "IsDuplicate": true,
            "Name": "Customer007 - Break",
            "RouteLocation": {
                "Latitude": 39.764119,
```

```
                "Longitude": -104.953665
            },
            "StopTimeMinutes": 30,
            "time_windows": null,
            "waiting_time": 0
        },
        {
            "Arrival": "11:47:19 12/16/2025",
            "Departure": "13:17:19 12/16/2025",
            "Distance": "2.05 miles (3.30 km)",
            "Duration": "6 minutes, 4 seconds",
            "IsDuplicate": false,
            "Name": "Customer004",
            "RouteLocation": {
                "Latitude": 39.750206,
                "Longitude": -104.933919
            },
            "StopTimeMinutes": 90,
            "time_windows": [
                {
                    "earliest": "09:00",
                    "latest": "12:00"
                },
                {
                    "earliest": "15:00",
                    "latest": "18:00"
                }
            ],
            "waiting_time": 0
        },
        {
            "Arrival": "13:32:42 12/16/2025",
            "Departure": "13:32:42 12/16/2025",
            "Distance": "5.54 miles (8.92 km)",
            "Duration": "15 minutes, 23 seconds",
            "IsDuplicate": false,
            "Name": "Depot",
            "RouteLocation": {
                "Latitude": 39.712797,
                "Longitude": -104.987282
            },
            "StopTimeMinutes": 0,
            "time_windows": null,
            "waiting_time": 0
        }
    ],
    "Route": {
        "DriveDistance": 28.612958,
```

```
                        "DriveDistanceUnit": "Kilometer",
                        "DriveTime": 2460,
                        "DriveTimeUnit": "Second",
                        "RouteLegs": [
                            {
                                "Directions": [
                                    "Continue onto East Cedar Avenue",
                                    "Turn left onto South Broadway",
                                    "Turn right onto West Alameda Avenue",
                                    "Turn right onto South Santa Fe Drive",
                                    "Keep left onto West Cedar Avenue",
                                    "Continue toward I 25 North",
                                    "Keep right and drive toward 23rd Avenue",
                                    "Keep right and drive toward Water Street,
Downtown, Aquarium, Children's Museum",
                                    "Continue onto Water Street",
                                    "Turn left onto 15th Street"
                                ],
                                "DriveDistance": 7.4512300000000016,
                                "DriveTime": 497,
                                "LegBegin": {
                                    "Name": "Depot",
                                    "RouteLocation": {
                                        "Latitude": 39.712797,
                                        "Longitude": -104.987282
                                    }
                                },
                                "LegEnd": {
                                    "Name": "Customer009",
                                    "RouteLocation": {
                                        "Latitude": 39.758191,
                                        "Longitude": -105.011556
                                    }
                                }
                            },
                            {
                                "Directions": [
                                    "Make a U-turn onto 15th Street",
                                    "Turn left onto Central Street",
                                    "Turn left toward I 25, US 6, US 85",
                                    "Keep right and take I 70 toward Limon, Grand
Junction",
                                    "Keep right and take I 70 East toward Limon",
                                    "Keep left",
                                    "Keep right onto US 6, US 85 and take US 6, US
85 toward Vasquez Boulevard",
                                    "Turn right onto North Steele Street",
                                    "Turn right onto East Bruce Randolph Avenue",
```

```
            "Turn left onto Fillmore Street"
        ],
        "DriveDistance": 8.940968,
        "DriveTime": 675,
        "LegBegin": {
            "Name": "Customer009",
            "RouteLocation": {
                "Latitude": 39.758191,
                "Longitude": -105.011556
            }
        },
        "LegEnd": {
            "Name": "Customer007",
            "RouteLocation": {
                "Latitude": 39.764119,
                "Longitude": -104.953665
            }
        }
      }, . . .
```


[API response shortened for documentation purposes due to length.]