RouteSavvy

It's your time and money. Make the most of it.

RouteSavvy Route Optimization API Documentation

Learn the technical highlights of the RouteSavvy Route Optimization API

- Programming tips
- GET & POST specifications
- Code samples

Contents

	Introduction	3
Section 1 -	What is the RouteSavvy API?	4
Section 2 -	GET, POST & URL Links for Developers	5
Section 2a -	Links to Code Samples	6
Section 3 -	Predictive Traffic Functions Sample Code	7
Section 4 -	OnTerra Systems USA Tech Support	25
Section 5 -	Free Trial & Purchase the RouteSavvy API	26

Introduction

For developers seeking an affordable, powerful route optimization engine to incorporate into applications being developed, the **RouteSavvy API** routing engine emerges as one of the most robust options on the market today. This versatile, flexible routing engine is written in a **RESTFUL/JSON** format, making it an easy & cost-effective option, as well as affordable. What's more, the RouteSavvy routing engine offers a wealth of options & benefits for developers, including:

- An affordable, flat-fee pricing structure
- Support for multiple programming languages
- 3 routing engine options
- Free code samples…and more

Check out the following documentation to learn more about the RouteSavvy route optimization API, how it works, how it saves time and money, and buying tips.

RouteSavvy API General & Purchase Information

https://www.routesavvy.com/routesavvy-route-optimization-api

FREE Trial

RouteSavvy API Free Trial Sign-up

https://www.routesavvy.com/routesavvy-api-free-trial

RouteSavvy API Online Documentation

https://www.routesavvy.com/routesavvy-api-docs

Section 1

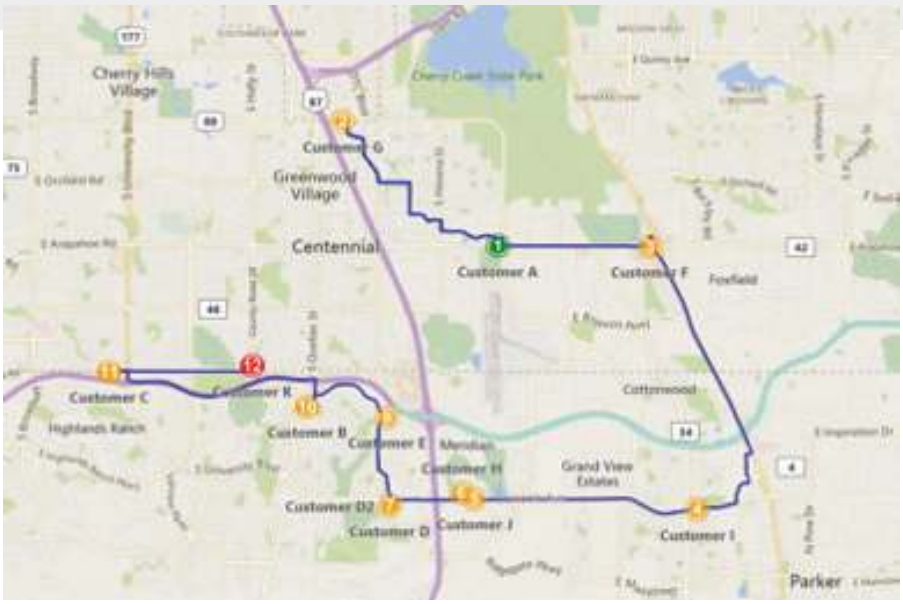
What Is the RouteSavvy API?

The **RouteSavvy Route Optimization API** is an easy-to-use REST service for optimizing driving stop order.

RSAPI.svc allows both GET and POST options.

Requests use JSON-formatted strings and results are returned in JSON format for optimal use in web applications.

The optimized results return stops ordered for the most efficient driving route. Results include turn-by-turn directions for each route leg and a latitude-longitude polyline describing the route for use in map visualization.



RouteSavvy Route Optimization API: This is a sample of optimized Stops and RoutePath displayed over Bing Maps.

Section 2

GET & POST Specifications and URL Links

GET Specifications

- RSAPI.svc/GetOptimize requests are limited to the URL-encoded parameter maximum length.
- Although this varies by browser, 2048 characters is the typical suggested limit for encoding stops in the URL string.
- Because of the URL length limitation smaller numbers of stops can be optimized.

POST Specifications

- RSAPI.svc/PostOptimize requests do not have URL length limitations and can accommodate large numbers of stops.

URLs

- **Base URL:** https://optimizer2.routesavvy.com
- **Service wsdl URL:** https://optimizer2.routesavvy.com/RSAPI.svc
- **GET URL:** https://optimizer2.routesavvy.com/RSAPI.svc/GetOptimize?query={jsonquery}
- **POST URL:** https://optimizer2.routesavvy.com/RSAPI.svc/PostOptimize

Section 2a

Links to Code Samples

You'll find code samples to view in this Documentation.

We also have actual code samples online for the following programming languages:

- C#
- Java
- JavaScript
- Python
- VB.NET

To obtain these actual code samples and download them, please visit: https://www.routesavvy.com/route-optimization-api-documentation/

Section 3

Predictive Traffic Functions & Sample Code

Predictive Traffic is an advanced route optimization feature that allows fleet managers to plan a multi-stop route based on when the route will be driven. This is accomplished because historical data on known traffic patterns (such as at rush hour) are incorporated into the calculations for the most efficient route.

Request JSON

Example JSON Request:


```
{
    "Locations": [{
        "Name": "Customer A",
        "Latitude": 39.595140,
        "Longitude": -104.849620,
        "VisitDurationInMinutes": 10
},
    {
        "Name": "Customer B",
        "Latitude": 39.558012,
        "Longitude": -104.906670,
        "VisitDurationInMinutes": 5
},
    {
        "Name": "Customer C",
        "Latitude": 39.566128,
```

Predictive Traffic Functions & Sample Code (continued)


```
            "Longitude": -104.965901,
            "VisitDurationInMinutes": 10
        },
        {
            "Name": "Customer D",
            "Latitude": 39.535220,
            "Longitude": -104.882080,
            "VisitDurationInMinutes": 10
        },
        {
            "Name": "Customer E",
            "Latitude": 39.555669,
            "Longitude": -104.882878,
            "VisitDurationInMinutes": 5
        },
    ],
    "OptimizeParameters": {
            "AppId": "90c13fb711be4684bc724d306321a609",
            "OptimizeType": "distance",
            "RouteType": "realroadcar",
            "Avoid": "none",
            "Departure": "2020-05-23T17:30:00-08:00"
        }
}
```

Predictive Traffic Functions & Sample Code (continued)

Request JSON

Request components:

Array of Locations – minimum 3 locations (note that no optimization is required until four or more stops are requested.)

	Example
{	
“Name”: string,	“Customer A”
“Latitude”: float,	39.595140
“Longitude”: float,	-104.906670
“VisitDurationInMinutes”: integer minutes	10
}	

Development Notes 1:

The first location is the start stop and the last location is the ending stop. If you wish to optimize a round-trip route, duplicate the start stop as the end stop.

Optimize Parameters:

“OptimizeParameters”: {	example:
“AppId”: string - user token	“90c13…..21a609”
“OptimizeType”: string - time or distance	“distance”
“RouteType”: string – basic ,realroadcar, realroadcarpredictive	“realroadcar”
“Avoid”: string - none, tolls, or highways	“none”
“Departure”: string datetime	“2020-05-23T17:30:00-08:00”

Predictive Traffic Functions & Sample Code (continued)

Development Notes 2:
RouteType basic is limited to 500 stops, realroadcar and realroadcarpredictive are limited to 300 stops.

RouteType realroadcarpredictive uses a future Departure date time for the start time, which accounts for historical variations by day of week, holidays, and rush-hour traffic.

Departure date time for realroadcarpredictive is date and time of departure from the origin point. The time zone will be assumed to be that of the origin latitude,longitude.

The departure value must be in the future in the date-time format: yyyy-MM-ddTHH:mm:ss

Example:
If Departure=2019-01-30T08:00:00 and the origin point is 39.595140, -104.906670 this will indicate a departure at 8:00am MST or 2019-01-30T15:00:00+00:00 UTC

The default departure behavior can be overridden by adding a timezone offset to the departure datetime. Example PST override 2019-01-30T08:00:00-08:00 yyyy-MM-ddTHH:mm:ss-HH:00.

Predictive Traffic Functions & Sample Code (continued)

Example:

Request:

Departure": "2020-01-23T08:00:00-08:00" will force time to PST and override timezone of origin location.

Result:

"Arrival": "2020-01-23T16:11:56+00:00",
"Departure": "2020-01-23T16:16:56+00:00"

RouteTypes, basic and realroadcar only use the departure time for labeling directions and calculating stop arrival/departure times but does not account for traffic variations. The depart date is always today's date and time zone offsets are ignored.

Here's a sample **GET Optimize** request which can run from a browser once your token has been pasted into the AppId.

http://optimizer2.routesavvy.com/RSAPI.svc/GETOptimize?query= { "Locations": [{ "Name": "Customer A", "Latitude": 39.595140, "Longitude": -104.849620, "VisitDurationInMinutes":10}, { "Name": "Customer B", "Latitude": 39.558012, "Longitude": -104.906670, "VisitDurationInMinutes":5}, { "Name": "Customer C", "Latitude": 39.566128, "Longitude": -104.965901, "VisitDurationInMinutes":10}, { "Name": "Customer D", "Latitude": 39.535220, "Longitude": -104.882080, "VisitDurationInMinutes":10}, { "Name": "Customer D2", "Latitude": 39.535220, "Longitude":

Predictive Traffic Functions & Sample Code (continued)

-104.882080, “VisitDurationInMinutes”:10},{ “Name”: “Customer E”, “Latitude”: 39.555669, “Longitude”: -104.882878, “VisitDurationInMinutes”:5}, { “Name”: “Customer F”, “Latitude”: 39.595030, “Longitude”: -104.804320, “VisitDurationInMinutes”:5}, { “Name”: “Customer G”, “Latitude”: 39.624000, “Longitude”: -104.896010, “VisitDurationInMinutes”:5}, { “Name”: “Customer H”, “Latitude”: 39.538212, “Longitude”: -104.860457, “VisitDurationInMinutes”:10}, { “Name”: “Customer I”, “Latitude”: 39.534298, “Longitude”: -104.790182, “VisitDurationInMinutes”:10}, { “Name”: “Customer J”, “Latitude”: 39.536745, “Longitude”: -104.856546, “VisitDurationInMinutes”:5}, { “Name”: “Customer K”, “Latitude”: 39.567508, “Longitude”: -104.923069, “VisitDurationInMinutes”:5}], “OptimizeParameters”: { “AppId”: “<user token>”, “OptimizeType”: “distance”, “RouteType”: “realroadcar”,”Avoid”: “none”, “Departure”: “2020-05-23T17:30:00”}}

Sample POST optimize request

http://optimizer2.routesavvy.com/RSAPI.svc/POSTOptimize

{
“Locations”: [{
“Name”: “Customer A”,
“Latitude”: 39.595140,
“Longitude”: -104.849620,
“VisitDurationInMinutes”: 10
},
{
“Name”: “Customer B”,

Predictive Traffic Functions & Sample Code (continued)

```
            "Latitude": 39.558012,
            "Longitude": -104.906670,
            "VisitDurationInMinutes": 5
        },
        {
            "Name": "Customer C",
            "Latitude": 39.566128,
            "Longitude": -104.965901,
            "VisitDurationInMinutes": 10
        },
{
            "Name": "Customer D",
            "Latitude": 39.535220,
            "Longitude": -104.882080,
            "VisitDurationInMinutes": 10
        },
        {
            "Name": "Customer E",
            "Latitude": 39.555669,
            "Longitude": -104.882878,
            "VisitDurationInMinutes": 5
        },
        "OptimizeParameters": {
```

Predictive Traffic Functions & Sample Code (continued)


```
            "AppId": "90c13fb711be4684bc724d306321a609",
            "OptimizeType": "distance",
            "RouteType": "realroadcar",
            "Avoid": "none",
            "Departure": "2020-05-23T17:30:00-08:00"
        }
}
```


Result JSON

Sample JSON Result:


```
{
        "Message": "Success",
        "OptimizedStops": [
            {
                "Arrival": null,
                "Departure": "9:25:00 3/29/2018",
                "Distance": null,
                "Duration": null,
                "IsDuplicate": false,
                "Name": "Customer A",
                "RouteLocation": {
                    "Latitude": 39.59514,
                    "Longitude": -104.84962
                },
                "StopTimeMinutes": 10
```

Predictive Traffic Functions & Sample Code (continued)


```
        },
        {
            "Arrival": "9:39:44 3/29/2018",
            "Departure": "9:44:44 3/29/2018",
            "Distance": "4.14 miles (6.67 km)",
            "Duration": "14 minutes, 44 seconds",
            "IsDuplicate": false,
            "Name": "Customer G",
            "RouteLocation": {
                "Latitude": 39.624,
                "Longitude": -104.89601
            },
            "StopTimeMinutes": 5
},
        {
            "Arrival": "10:06:40 3/29/2018",
            "Departure": "10:11:40 3/29/2018",
            "Distance": "6.88 miles (11.07 km)",
            "Duration": "21 minutes, 56 seconds",
            "IsDuplicate": false,
            "Name": "Customer F",
            "RouteLocation": {
                "Latitude": 39.59503,
                "Longitude": -104.80432
            },
```

Predictive Traffic Functions & Sample Code (continued)


```
                "StopTimeMinutes": 5
            },
                .
                .
                .
            },
{
                "Arrival": "12:03:39 3/29/2018",
                "Departure": "12:08:39 3/29/2018",
                "Distance": "2.38 miles (3.84 km)",
                "Duration": "4 minutes, 59 seconds",
                "IsDuplicate": false,
                "Name": "Customer K",
                "RouteLocation": {
                    "Latitude": 39.567508,
                    "Longitude": -104.923069
                },
                "StopTimeMinutes": 5
            }
        ],
        "Route": {
            "DriveDistance": 51.698,
            "DriveDistanceUnit": "Kilometer",
            "DriveTime": 5019,
            "DriveTimeUnit": "Second",
            "RouteLegs": [
```

Predictive Traffic Functions & Sample Code (continued)


```
{
    "Directions": [
        "Depart CO-88 W / E Arapahoe Rd
toward S Paris St",
        "Turn right onto S Paris St",
        "Turn left onto E Peakview Ave",
        "Turn right onto S Havana St",
        "Turn left onto E Caley Ave",
        "Turn right onto S Dayton St",
        "Turn left onto E Orchard Rd",
        "Turn right onto DTC Blvd",
        "Turn left onto E Prentice Ave",
        "Bear right onto S Ulster St",
        "Turn left onto E Belleview Ave",
        "Make a U-turn at Promenade Pl",
        "Arrive at E Belleview Ave"
    ],
    "DriveDistance": 6.668,
    "DriveTime": 884,
    "LegBegin": {
        "Name": "Customer A",
        "RouteLocation": {
            "Latitude": 39.59514,
            "Longitude": -104.84962
        }
    },
```

Predictive Traffic Functions & Sample Code (continued)


```
            "LegEnd": {
                "Name": "Customer G",
                "RouteLocation": {
                    "Latitude": 39.624,
                    "Longitude": -104.89601
                }
            }
        },
        {
                "Directions": [
                "Depart E Belleview Ave toward S Ulster St",
                "Turn right onto S Ulster St",
                "Turn left onto E Prentice Ave",
                "Turn right onto DTC Blvd",
                "Turn left onto E Orchard Rd",
                "Turn right onto S Dayton St",
                "Turn left onto E Caley Ave",
                "Turn right onto S Havana St",
                "Turn left onto E Peakview Ave",
                "At roundabout, take 1st exit onto S Peoria St",
                "Turn left onto CO-88 / E Arapahoe Rd",
                "Turn left onto S Helena St",
                "Turn right onto road",
                "Take ramp right",
                "Arrive at ramp"
```

Predictive Traffic Functions & Sample Code (continued)


```
        ],
        "DriveDistance": 11.065,
        "DriveTime": 1316,
        "LegBegin": {
            "Name": "Customer G",
            "RouteLocation": {
                "Latitude": 39.624,
                "Longitude": -104.89601
            }
        },
        "LegEnd": {
            "Name": "Customer F",
            "RouteLocation": {
                "Latitude": 39.59503,
                "Longitude": -104.8043
            }
        }
    },
        .
        .
        .
    {
        "Directions": [
            "Depart E County Line Rd toward E County Line Rd",
            "Turn left onto County Road 29 / S Holly St",
            "Arrive at County Road 29 / S Holly St"
```

Predictive Traffic Functions & Sample Code (continued)


```
                    ],
                    "DriveDistance": 3.838,
                    "DriveTime": 299,
                    "LegBegin": {
                        "Name": "Customer C",
                        "RouteLocation": {
                            "Latitude": 39.566128,
                            "Longitude": -104.965901
                        }
                    },
                    "LegEnd": {
                        "Name": "Customer K",
"RouteLocation": {
                            "Latitude": 39.56745,
                            "Longitude": -104.92307
                        }
                    }
                }
            ],
            "RoutePath": [
                [39.595139,-104.849619],
                [39.595139,-104.85009],
                [39.59552,-104.85009],
                    .
                    .
                    .
```

Predictive Traffic Functions & Sample Code (continued)

```
                [39.565979,-104.92306],
                [39.56614,-104.923069],
                [39.56745,-104.923069]
            ]
        }
}
```

Result Components

```
{
    "Message": string - "Success" or error message
    "OptimizedStops": Array of stops
    "Route": route details
}
```

OptimizedStops Array of stops in the new optimized order

Stops	Example
"Arrival": string date time	"9:39:44 3/29/2018",
"Departure": string date time	"9:44:44 3/29/2018",
"Distance": string distance mi and km	"4.14 miles (6.67 km)",
"Duration": string drive time min and sec	"14 minutes, 44 seconds",
"IsDuplicate": Boolean	false,
"Name": string	"Customer G",
"RouteLocation": {	
"Latitude": float	39.624,
"Longitude": float	-104.89601
},	
"StopTimeMinutes": integer	5

OptimizedStops

Predictive Traffic Functions & Sample Code (continued)


```
[
        {
            "Arrival": null,
            "Departure": "9:25:00 3/29/2018",
            "Distance": null,
            "Duration": null,
            "IsDuplicate": false,
            "Name": "Customer A",
            "RouteLocation": {
                "Latitude": 39.59514,
                "Longitude": -104.84962
            },
            "StopTimeMinutes": 10
        },
        {
            "Arrival": "9:39:44 3/29/2018",
            "Departure": "9:44:44 3/29/2018",
            "Distance": "4.14 miles (6.67 km)",
            "Duration": "14 minutes, 44 seconds",
            "IsDuplicate": false,
            "Name": "Customer G",
            "RouteLocation": {
                "Latitude": 39.624,
                "Longitude": -104.89601
            },
            "StopTimeMinutes": 5
        },
```

Predictive Traffic Functions & Sample Code (continued)

Result Components

```
{
        "DriveDistance": float total distance in DriveDistanceUnit    51.698,
        "DriveDistanceUnit": string                                   "Kilometer",
        "DriveTime": integer time in DriveTimeUnit                    5019,
        "DriveTimeUnit": string                                       "Second",
        "RouteLegs": Array of Route Legs                              [...]
        "RoutePath": Array of lat,lon                                 [...]
    }
```

RouteLegs

```
{                                   Example
    "Directions": Array of strings
                                    [
                                        "Depart CO-88 W toward S Paris St",
                                        "Turn right onto S Paris St",
                                        "Turn left onto E Peakview Ave",
                                        "Turn right onto S Havana St",
                                        "Turn left onto E Caley Ave",
                                        "Turn right onto S Dayton St",
                                        "Turn left onto E Orchard Rd",
],
    "DriveDistance": float in DriveDistanceUnit 6.668,
    "DriveTime": integer in DriveTimeUnit       884,
    "LegBegin": {
        "Name": string                  "Customer A",
        "RouteLocation": {
            "Latitude": float           39.59514,
            "Longitude": float          -104.84962
```

Predictive Traffic Functions & Sample Code (continued)

```
            }
},
        "LegEnd": {
            "Name": string                  "Customer G",
            "RouteLocation": {
                "Latitude": float           39.624,
                "Longitude": float          -104.89601
                }
            }
        },
```

RoutePath

```
[
    [ 39.595139,-104.849619],
    [ 39.595139,-104.85009],
    [ 39.59552,-104.85009],
            .
            .
            .
```

Errors results

If there is an error, Message returns information about the error. OptimizedStops and Route are null.

```
{
"Message": "Invalid Token. For assistance, please contact routesavvy@onterrasystems.com",
            "OptimizedStops": null,
        "Route": null
```

Section 4

OnTerra Systems Tech Support

OnTerra Systems USA offers exceptional tech support for developers who need to incorporate route optimization into software solutions they are developing.

How to Contact OnTerra Systems for Tech Support

For tech support on incorporating the RouteSavvy API into your application or software:

support@routesavvy.com

Section 5

Free Trial or Purchase the RouteSavvy API

RouteSavvy API General & Purchase Information

https://www.routesavvy.com/routesavvy-route-optimization-api

FREE

RouteSavvy API Free Trial Sign-up

https://www.routesavvy.com/routesavvy-api-free-trial

RouteSavvy API Online Documentation

https://www.routesavvy.com/routesavvy-api-docs