

# REST Service: Partitioner

[ROUTEPERFORM.COM](https://routeperform.com)

ROUTE PLANNING WEB SERVICES FOR DEVELOPERS

# WELCOME

---

WELCOME TO THE PARTITIONER WEB SERVICE REFERENCE GUIDE. THIS WEB SERVICE MAKES A VERY DIFFICULT MATHEMATICAL CHALLENGE SEEMS QUITE SIMPLE. YOU CAN PROVIDE A LARGE NUMBER OF GEOGRAPHIC POINTS (FOR INSTANCE 100 OR 1,000+) AND ALSO PROVIDE A SPECIFIC NUMBER OF PARTITIONS TO BE CREATED AND THE SERVICE WILL RETURN TO YOU GEOGRAPHIC PARTITIONS (ALSO KNOWN AS 'CLUSTERS').

THE TOOL CAN BE USED FOR BOTH STRATEGIC AND TACTICAL PURPOSES. FOR INSTANCE, YOU COULD TAKE ALL 1,000 CUSTOMERS AND DECIDE TO CREATE FOUR GEOGRAPHIC PARTITIONS WITH SIMILAR/EQUAL REVENUE AS A STRATEGIC PLANNING EXERCISE. OR FOR A MORE TACTICAL USE CASE YOU COULD PROVIDE 200 ROUTE STOPS FOR TODAY'S TACTICAL PURPOSES AND ASK THE SERVICE FOR 10 ROUTES OF EXACTLY 20 STOPS EACH TO BE RETURNED. THESE ARE JUST 2 EXAMPLES, BUT THERE ARE A VARIETY OF WAYS THAT THIS SERVICE CAN BE USED TO APPLY HELPFUL LOGIC TO YOUR LOGISTICAL OPERATIONS.

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 PARTITIONER CALL IS A RESTFUL WEB SERVICE. WE USE JSON AS INPUT AND OUTPUT.

# SERVICE OVERVIEW & PURPOSE

---

SOME KEY POINTS TO KNOW ABOUT THE PARTITIONER WEB SERVICE:

- A STRONG-POINT OF THE PARTITIONER SERVICE IS THAT YOU CAN SEND A LARGE AMOUNT OF STOPS TO THE SERVICE AND IT CAN RETURN THE RESULT TYPICALLY WITHIN SECONDS. IT IS THEREFORE A GREAT WAY TO TAKE A MASS OF DATA AND TO BREAK IT INTO GEOGRAPHICALLY DISTINCT PARTITIONS (AKA 'CLUSTERS'). THE WAY IT CAN HANDLE THIS VOLUME OF DATA SO QUICKLY IS TO LOOK AT THE 'BIG PICTURE' AND TO APPLY ALGORITHMS AND HEURISTICS TO GENERATE AN ANSWER FOR YOU. IT IS NOT GOING TO RETURN SEQUENCED RESULTS OR MEASUREMENTS ON TIMES IT WOULD TAKE TO DRIVE TO THE POINTS, YOU'D WANT TO USE OUR OTHER WEB SERVICES FOR THOSE GRANULAR BITS. THIS WEB SERVICE IS TYPICALLY BEST USED AS A FIRST-PASS OR STRATEGICAL PROCESS.
- THE SERVICE ALLOWS FOR UP TO 1000 STOPS PER REQUEST
- THE SERVICE PURPOSEFULLY RETURNS PARTITIONED RESULTS FOR THE SET OF PARTITION TEMPLATES THAT YOU PROVIDE. IN OTHER WORDS, IF YOU WISH TO BREAK 1,000 STOPS INTO 3 GROUPS THEN YOU PASS IT 3 TEMPLATE PARTITIONS AS AN INPUT AND IT ABIDES BY THAT INPUT QUITE STRICTLY. EACH PARTITION TEMPLATE CAN HAVE ITS OWN INPUT COORDINATE SO THAT YOU CAN INTELLIGENTLY GIVE FOR INSTANCE WESTERN GROUPS OF STOPS TO TEMPLATES SET ON A WESTERN COORDINATE THAT YOU'VE PROVIDED. YOU CAN ALSO PROVIDE MULTIPLE PARTITION TEMPLATES WITH COMMON/IDENTICAL GEOGRAPHIC POINTS TOO SUCH AS IF 10 VEHICLES WERE ALL GOING TO START/END AT A COMMON DEPOT.
- THE SERVICE ALLOWS YOU TO SPECIFY HOW YOU WISH TO BALANCE THE WORK SUCH AS BY TIME, BY COUNT, OR BY A NUMERIC FACTOR THAT YOU SPECIFY (REVENUE OR WEIGHT OR CUBES OR POINTS, ETC.)

# ABOUT PARTITIONING

---

WE'D BE REMISS TO NOT GIVE YOU A LITTLE OVERVIEW ON GEOGRAPHIC PLANNING ITSELF. IT TURNS OUT THAT THE MATH REQUIRED TO ACCOMPLISH PARTITIONING/CLUSTERING IS REALLY HARD. IN FACT, IT IS MONUMENTALLY HARD. IT IS WHAT THEY CALL IN THE MATH WORLD AN 'N-HARD' CHALLENGE. AS IN, YOU CANNOT SOLVE IT WITH BRUTE FORCE, YOU NEED BRAINS APPLIED AS BRAWN ALONE ISN'T GOING TO CUT IT.

AS AN EXAMPLE:

3 STOPS = 6 POSSIBLE COMBINATIONS (ABC, ACB, BAC, BCA, CAB, CBA)

6 STOPS = 720 POSSIBLE COMBINATIONS

11 STOPS = 31 MILLION+ COMBINATIONS

20 STOPS = 2,432,000,000,000,000,000+ COMBINATIONS (WHOA...)

100 STOPS =  $9.332621544 \times 10^{157}$  - AND YEP, WE STILL CAN SOLVE IT

FRET NOT! IF YOU ARE READING THIS DOCUMENT THEN YOU'VE FOUND THE RIGHT TOOL FOR THE JOB. WE'VE TUNED OUR SERVICE AND ALGORITHMS OVER THE COURSE OF TIME BY BUILDING TENS OF THOUSANDS OF PARTITIONS FOR OUR CLIENTS.

AS YOU CODE YOUR SIMPLE REQUESTS TO OUR PARTITIONING SERVICE YOU GET TO BE THE INSTANT BENEFICIARY OF INDUSTRY-PROVEN LOGIC THAT WILL ALLOW YOU TO EASILY ADD THIS FUNCTIONALITY TO YOUR EXISTING CORE SOFTWARE. OUR ARCHITECTURE RUNS IN THE AWS AND AZURE CLOUD TO ENSURE THE BEST POSSIBLE SCALABILITY AND RELIABILITY FOR USERS OF THIS SERVICE.

# 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 SYNCHRONOUSLY. AS FOR THE REQUEST SYNTAX WE REQUIRE, EVERYTHING IS OUTLINED BELOW.

## ↔ Sample Code: Request URI

<https://www.routeperform.com/services/v1/partitioner>

see below for information on body parameters

# 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](https://en.wikipedia.org/wiki/JSON)

## ↔ Sample Request - Example

```
{
  "apiKey": "{yourKey}",
  "passthroughGUID": "{GUID}",
  "requestOptions": {
    "balanceByFactor": 1,
    "keepCommonPointsTogether": true
  },
  "inputPartitionTemplates": [
    {
      "inputPartitionTemplateID": "801",
      "inputPartitionTemplateName": "Joe Smith"
      "latitudeY": -82.392392,
      "longitudeX": 32.393921,
    },
    { etc...
  }
],
  "inputStops": [
    {
      "stopID": "801",
      "latitudeY": -82.1212,
      "longitudeX": 32.6767,
      "serviceMinutes": "5",
      "numericFactor": "100"
    },
    {
      "stopID": "802",
      "latitudeY": -82.392392,
      "longitudeX": 32.393921,
      "serviceMinutes": "10",
      "numericFactor": "200"
    }
  ]
}
```

## 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).

## REQUEST OPTIONS - PARAMETERS

Parameter name	Type	Description
balanceByFactor	integer	Valid values: <ul style="list-style-type: none"><li>• 1 - Time</li><li>• 2 - Distance</li><li>• 3 - StopCount</li><li>• 4 - NumericFactor</li></ul> Note: 1-Time is the default, however if the problem size is sufficiently large (wide geography coupled with very high point count) then it can be auto-set to 3-StopCount in certain cases to prevent processing times that are too lengthy.
keepCommonPointsTogether	boolean	If 'true', then points in common (that share latitude/longitude exactly) are always kept together. The benefit to keeping them together is to avoid the possibility that common stops are split-up to favor balance over optimization. The detriment to keeping them together is that it can negatively alter the balance of the returned solution in favor of optimization. 'true' is the default and recommended option for most data sets. Large datasets may flip the setting to 'false' internally in order to keep processing times in check as the grouping phase is computationally expensive.



## INPUTPARTITIONTEMPLATE – PARAMETERS



Note: Input partition templates allow you to dictate how many partitions will be created. For instance, if you provide 3 templates as inputs, then 3 templates will be returned as a result. Each partition template must be given a latitude/longitude that acts as a geographic centerpoint (a magnetic center, so to speak) for the partition that will be created. That can be a real position, such as a depot yard, or can be a dynamically chosen coordinate such as point that generally handles the western point of your service area. These centerpoints can be unique from one another, or can be identical to one another, and any combination of the two.

Note: No more than 50 input partition templates should be supplied.

Parameter name	Type	Description
inputPartitionTemplateID	string (50)	A unique identifier for an input partition template. A value must be supplied, and values may not contain spaces. Use of only alpha and numeric characters is encouraged.
inputPartitionTemplateName	string (50)	A human-readable name for an input partition template such as 'route1' or 'west region', etc. If left blank, the 'id' value supplied will be applied.
latitudeY	double	(required) Latitude portion of the geographic coordinate. Example: 32.708328
longitudeX	double	(required) Longitude portion of the geographic coordinate. Example: -117.161133

## INPUTSTOPS- PARAMETERS



Note: No more than 1,000 stops should be supplied.

Parameter name	Type	Description
stopID	String (50)	A unique identifier for an input stop. A value must be supplied, and values may not contain spaces. Use of only alpha and numeric characters is encouraged.
latitudeY	double	(required) Latitude portion of the geographic coordinate. Example: 32.708328
longitudeX	double	(required) Longitude portion of the geographic coordinate. Example: -117.161133
serviceMinutes	double	Optional. The number of minutes to service the stop upon arrival. Must be non-negative and less than 999 minutes per stop. This is only evaluated if balanceByFactor = time.
numericFactor	double	Optional. Used to specify a numeric factor to evaluate as a means to balance the partitions created. Examples could be weights, volumes, points, pallets, cases or currency. This is only evaluated if balancyByFactor = numericFactor. Values that are negative or zero will be set to = 1. Individual values may not exceed 9,999.

# RESPONSES – 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](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](https://en.wikipedia.org/wiki/JSON)

THE RESULTCODE VALUE WILL VERIFY THE SUCCESS OR FAILURE OF THE REQUEST. APPENDIX A LISTS ALL POSSIBLE RESULT CODES.

## Sample Response

```
⏪ {
  "passThroughGUID": "{GUIDfromRequest}",
  "outcome": {
    "resultCode": 1,
    "resultCodeDesc": "SuccessfullyRoutedAllItems"
  },
  "outputPartitions": [
    {
      "partitionID": "123",
      "outputStopsTotalCount": 0,
      "outputStopsTotalNumericFactor": 0,
      "partitionStops": [
        {
          "stopID": "xyz",
          "latitudeY": 32.728328,
          "longitudeX": -117.171133
        },
        {
          "stopID": "abc",
          "latitudeY": 32.929292,
          "longitudeX": -117.123123
        }
      ]
    },
    {
      ...next outputPartition, etc.
    }
  ]
}
```

# 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.

## RESPONSE-OUTCOME

Element name	Type	Description
resultCode	integer	See Appendix A for full information.
resultCodeDesc	string	Text that is a readable representation of the result code returned.
resultBalanceByFactor	string	This supplies the response with the passthrough input of the 'balanceByFactor' value.

## RESPONSE-OUTPUTPARTITIONS

Element name	Type	Description
partitionID	string (50)	
outputStopsTotalCount	double	
outputStopsTotalNumericFactor	double	
partitionStops	double	(see section below)

## RESPONSE-PARTITIONSTOPS

Element name	Type	Description
stopID	string	
latitudeY	double	Latitude portion of the geographic coordinate that was supplied with the request. Example: 32.708328
longitudeX	double	Longitude portion of the geographic coordinate that was supplied with the request. Example: -117.161133

## RESPONSE - OUTPUTMESSAGES



Note: Messages are returned in an array (if any) and may exist for notes, warnings and errors.

Element name	Type	Description
messageType	string	
messageText	string	

# APPENDIX A – RESULT CODES

---

## RESULTCODES

Result Code	Value	Notes
SuccessfullyProcessed	2000	All items were partitioned. Not to be confused with the http result code of 200.
SuccessfullyProcessedPartial	2001	A partitioned result was returned but there are one or more items not processed or serious messages to note.
NotProcessed	0	
ErrorNoAPIKeySupplied	1	
ErrorInvalidAPIKeySupplied	2	
ErrorInvalidRequestSupplied	10	Please verify your JSON data is in valid format.
ErrorDuringPreValidation	20	General error when validating input data.
ErrorNoInputPartitionTemplatesSupplied	1000	
ErrorTooManyPartitionTemplatesSupplied	1001	
ErrorZeroLengthPartitionTemplateID	1002	
ErrorTooLengthyPartitionTemplateID	1003	
ErrorPartitionTemplateIDContainedIllegalCharacter	1004	Disallowed: Pipe ( )
ErrorNoInputStopsSupplied	1100	
ErrorTooManyInputStopsSupplied	1101	
ErrorZeroLengthStopID	1102	
ErrorTooLengthyStopID	1103	

ErrorStopIDContainedIllegalCharacter	1104	Disallowed: Pipe ( )
ErrorStopIDValueProvidedWasNotUnique	1105	
ErrorInvalidLatLonValueProvidedForStop	1106	
ErrorDuringProcessing	1200	
ErrorDuringReturnValueHandling	1201	