V2

# REST Service: Point in Polygon

ROUTEPERFORM.COM

ROUTE PLANNING WEB SERVICES FOR DEVELOPERS

# Welcome

Welcome to the 'Point in Polygon' web service reference guide. This web service allows you to pass in a set of points and a set of polygons and it will return complete information related to the geographic relation of each point to each polygon. In other words, the service can help you to identify which points (if any) fall within the geographic scope of each polygon. This may sound easy, but polygons can be quite complex and points can fall within zero polygons or one polygon or multiple polygons. Our service will not only supply you with complete lookup results for all input data but also will provide additional context such as the 'best' polygon match for each point regardless of 0, 1 or 2+ polygons pass the 'Point in Polygon' lookup.

The practical uses of the tool are numerous.

As one example, you may possess known geographic zones (as polygons) for route drivers, such as 4 drivers that each have a drawn quadrant that identifies their geographic service Region. You can then provide this service with those zone polygons as well as the daily points, such as their work orders for the day, as input. The service output then will clearly specify the polygon(s) that each point was located within. Therefore, you can assign work orders to drivers via geography with no ongoing human involvement.

Other usage examples are numerous, such as testing points within the geographic bounds of time zones, states, counties, zip codes, etc.

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 points and polygons to be geocoded (possess lat/lon coordinates). If your data items 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 – Worldwide

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

rest/json - The 'Point in Polygon' call is a RESTful web service. We use JSON as input and output.

# 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/point-in-polygon

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

### Sample Request - Example

```
{
    "apiKey": "{yourKey}",
    "passthroughGUID": "{GUID}",
    "requestOptions": {
      "doOutputPolygonGeometry": true
    },
    "inputPoints": [
       {
         "pointID": "1",
         "pointName": "Work Order 19203",
         "latitudeY": 32.393921,
         "longitudeX": -117.29184
       },
       { etc…
       }
    ],
    "inputPolygons": [
       {
         "polygonID": "A",
         "polygonName": "XYZ Zone",
         "points": [
           {
             "latitudeY": 32.6767,
             "longitudeX": "-117.019123"
           }, etc…
         ]
       }, etc…
    ]
}
```

## 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 |
|---|---|---|
| doOutputPolygonGeometry | boolean | If 'true', then the points that comprise the polygons input are provided in the output. |

## inputPoints – Parameters

Note: No more than 1000 input points can be supplied.

| Parameter name | Type | Description |
|---|---|---|
| pointID | string (50) | A unique identifier for an input point. A value must be supplied, and values may not contain spaces or commas. Use of only alpha and numeric characters is encouraged. |
| pointName | string (50) | A human-readable name for an input point. |
| latitudeY | double | (required) Latitude portion of the geographic coordinate. Example: 32.708328 |
| longitudeX | double | (required) Longitude portion of the geographic coordinate. Example: -117.161133 |

### inputPolygons– Parameters

Note: No more than 100 polygons can be supplied.

| Parameter name | Type | Description |
|---|---|---|
| polygonID | string (50) | A unique identifier for an input polygon. A value must be supplied, and values may not contain spaces or commas. Use of only alpha and numeric characters is encouraged. |
| polygonName | string (50) | A human-readable name for an input polygon. |
| points | array | Each polygon must be comprised of 4 or more geometric points that comprise the polygon. First and last supplied points must match to close-out the polygon. See 'points' area for more information. |

### points– Parameters

Note: No more than 500 geometric points can be supplied per polygon.

| Parameter name | Type | Description |
|---|---|---|
| latitudeY | double | (required) Latitude portion of the geographic coordinate. Example: 32.708328 |
| longitudeX | double | (required) Longitude portion of the geographic coordinate. Example: -117.161133 |

# Responses – Overview

Review the HttpStatusCode first. Value 200 'OK' confirms the service returned a response but does not necessarily verify that the point in polygon results could be returned. In the event such as bad input data it could for instance return a 200 'OK' but lack functional 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 A lists all possible result codes.

## Sample Response


```
{
    "passThroughGUID": "{GUIDfromRequest}",
    "outcome": {
      "resultCode": 2000,
      "resultCodeDesc": "Successfully processed"
    },
    "outputPolygons": [
       {
          "polygonID": "XYZ",
          "polygonName": "City Airport",
          "centerOfExtentsLatitudeY": 32.728328,
          "centerOfExtentsLongitudeX": -117.171133,
          "countOfPointsWithin": 0,
          "pointIDsWithin": "1,2,9"
       },
       {
          …etc…
       }
    ],
    "outputPoints": [
       {
          "pointID": "1992",
          "pointName": "NK Local School",
          "latitudeY": 32.92822,
          "longitudeX": -117.992113,
          "countOfPolygonsWithin": 1,
          "polygonIDsWithin": "RQZ",
          "closestPolygonIDByCenterOfExtents": "RQZ"
       },
       {
          …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. |

## RESPONSE-OUTPUTPOLYGONS

| Element name | Type | Description |
|---|---|---|
| polygonID | string (50) | |
| polygonName | string (50) | |
| centerOfExtentsLatitudeY | double | A coordinate returned as a courtesy that represents one method of approximating the centerpoint of a polygon. Latitude portion of the geographic coordinate. Example: 32.708328 |
| centerOfExtentsLongitudeX | double | A coordinate returned as a courtesy that represents one method of approximating the centerpoint of a polygon. Longitude portion of the geographic coordinate. Example: -117.161133 |
| countOfPointsWithin | double | A count of the number of input points that were geographically within the bounds of this polygon. |
| pointIDsWithin | double | A comma delimited list of input pointID values that were geographically within the bounds of this polygon. |

## Response-outputPoints

| Element name | Type | Description |
|---|---|---|
| pointID | string (50) | |
| pointName | string (50) | |
| 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 |
| countOfPolygonsWithin | integer | The count of polygons that this input point was geographically within. |
| polygonIDsWithin | string | A comma delimited list of input polygonID values that this input point was geographically within. |
| closestPolygonIDByCenterOfExtents | string | The poylgonID value of the polygon most geographically aligned with this input point. A point need not be within any polygon to have a value returned. If a point is within multiple polygons then the polygonID of the polygon with the centerpoint most similar to this input point is returned. |

## 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 processed. Not to be confused with the http result code of 200. |
| SuccessfullyProcessedPartial | 2001 | A 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. |
| ErrorDuringProcessing | 30 | General error while processing input data. |
| ErrorNoInputPointsSupplied | 1100 | |
| ErrorTooManyInputPointsSupplied | 1101 | |
| ErrorZeroLengthPointID | 1102 | |
| ErrorTooLengthyPointID | 1103 | |
| ErrorPointIDContainedIllegalCharacter | 1104 | Disallowed: Pipe (‘|’, ‘,’) |
| ErrorPointIDValueProvidedWasNotUnique | 1105 | |
| ErrorInvalidLatLonValueProvidedForPoint | 1106 | |
| ErrorNoInputPolygonsSupplied | 1200 | |
| ErrorTooManyInputPolygonsSupplied | 1201 | |
| ErrorZeroLengthPolygonID | 1202 | |
| ErrorTooLengthyPolygonID | 1203 | |
| ErrorPolygonIDContainedIllegalCharacter | 1204 | Disallowed: Pipe (‘|’, ‘,’) |
| ErrorPolygonIDValueProvidedWasNotUnique | 1205 | |
| ErrorPolygonHadZeroPoints | 1210 | |
| ErrorPolygonHadTooFewPoints | 1211 | |
| ErrorPolygonHadTooManyPoints | 1212 | |
| ErrorInvalidLatLonValueProvidedForVertexPoint | 1213 | |
| ErrorPolygonFirstAndLastPointsDontMatch | 1214 | |