# Routing API Mapping Documentation [en]

- **bold** = Default Value
- *italic* = function
- ***bold_italic*** = mapping not possible
- `[i] =` Given JSONObject in the JSONArray
- `< ... >` = final property that holds value(s)
- "Request Parameter" = data that is passed along in a request to the proxy, but not used by Graphhopper directly. This information is only used in the mappings. Serves the purpose of filling holes of information that is not included in a Graphhopper response

#### Mapbox Optimization Response Object

| Mapbox | needed type | Used Graphhopper Data | type | Conversion | Comment |
|---|---|---|---|---|---|
| `routes` | Array<RouteObject> | | | | |
| `waypoints` | Array<WaypointObject> | | | | |
| `code` | String | | | **Ok** | |
| `uuid` | String | | | *getUuid()*<br><br>generates unique UUID | The usage for this is not clear on Mapbox's side. Seems to have internal purpose |

#### Waypoint Object

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| name | String | `paths[i].instructions[i].<street_name>`<br><br>(here `i` = either first or last instruction) | String | *getFirstStreetName()*<br><br>*getLastStreetName()* | |
| `location` | Array<Double> | `paths[i].snapped_waypoints.<coordinates>` | Double | No conversion needed | |

#### Route Object

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| `distance` | Integer | `paths[i].<distance>` | Integer | No conversion needed | |
| `duration` | Integer | `paths[i].<duration>` | Integer | divide by 1000 | ms s |
| `geometry` | String (polyline) | `paths[i].points.<coordinates>` | Array<Double> | *polyline.encode()*<br><br>returns polyline encoded String of Array | |
| `weight` | Integer | (See `.duration`) | Integer | | It is not clear if the weight property is actually important |
| `weight_name` | String | | | **routability** | |
| `legs` | Array<LegObject> | | | | |
| `routeOptions` | RouteOptions Object | | | | |
| `voiceLocale` | String | Request Parameter: `locale` | String | No conversion needed | |

#### Leg Object

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| `distance` | Integer | (See Route Object `distance`) | | | As of now, a route contains of only one leg, since only A B navigation is supported. A B C navigation is not. |
| `duration` | Integer | (See Route Object `duration`) | | | " |
| `summary` | String | (See Waypoint Object `name`)<br><br>(First and Last Street Name) | | *getSummary()*<br><br>returns "A.street_name to B. street_name"<br><br>*getFirstStreetName()*<br><br>*getLastStreetName()* | |
| `steps` | Array<StepObject> | | | | |

**Step Object**

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| `name` | String | `paths[i].instructions[i].<street_name>` | String | No conversion needed | |
| `duration` | Integer | `paths[i].instructions[i].<time>` | Integer | <time> / 1000 | ms s |
| `weight` | Integer | (See `.duration`) | | | |
| `distance` | Integer | `paths[i].instructions[i].<distance>` | Integer | No conversion needed | |
| `geometry` | String (polyline) | `paths[i].instructions[i].<interval>`<br><br>`paths[i].points.<coordinates>` | | *polyline.encode( sectionPoints)*<br><br>Returns an encoded polyline String of those points that are part of the instruction interval | `interval` saves the indexes for the `points` array, to look up the coordinates which part of an instruction. |
| `driving_side` | String | ***No way to get the driving_side, as it is not part of the GH response*** | String | **right** | Workaround would be to detect the country that your navigating in and by that decide the driving_side |
| `mode` | String | Request parameter: `vehicle` | String | *convertProfile()*<br><br>returns the mode of vehicle (e.g car driving, foot walking) | |
| `maneuever` | ManeuverObject | | | | |
| `intersections` | Array <IntersectionObject> | | | | |
| `voiceInstructions` | Array <VoiceInstructionsObject> | | | | |
| `bannerInstructions` | Array <BannerInstructionsObject> | | | | |

**Maneuver Object**

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| `bearing_before` | Integer | `paths[i].instructions[i].<interval>`<br><br>`paths[i].points.<coordinates>` | Array<Integer><br><br>Array<Double> | *getBearingBefore()*<br><br>returns the bearing of the vehicle before a maneuver (= angle between either the coordinates of last intersection or location of current instruction and location of the maneuver) | 0 <= bearing <= 360°<br>0° north<br><br>180° south |

| | | | | | |
|---|---|---|---|---|---|
| `bearing_after` | Integer | `paths[i].instructions[i+1].<interval>`<br><br>`paths[i].points.<coordinates>` | Array<Integer><br><br>Array<Double> | *getBearingAfter()*<br><br>returns the bearing of the vehicle after a maneuver (= angle between location of maneuver and either the coordinates of first intersection or coordinates of next maneuver) | 0 >= bearing <= 360°<br>0° north<br><br>180° south |
| `location` | Array<Double> | `paths[i].instructions[i].<interval>`<br><br>`paths[i].points.<coordinates>` | Array<Integer><br><br>Array<Double> | the first index of `<interval>` used as lookup for `<coordinates>` | |
| `modifier` | String | `paths[i].instructions[i].<sign>` | Integer | *getMapboxModifier()*<br><br>converts the Graphhopper `<sign>` integers to `modifier` like "sharp left", "right" etc. | |
| `type` | String | `paths[i].instructions[i].<sign>` | Integer | *getType()*<br><br>converts the Graphhopper `<sign>` Integers to `type` s like "arrive", "rounabout" etc. | |
| `instruction` | String | `paths[i].instructions[i].<text>` | String | No conversion needed | |
| `exit`<br><br>*(option for* `type=roundabout` *)* | Integer | `paths[i].instructions[i].<exit_number>` | Integer | No conversion needed | Number of the exit in a roundabout |

**Intersection Object**

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| `out` | Integer | | | **0** | |
| `entry` | Array<Boolean> | | | **[true]** | See mapbox Doc for Intersection explanation.<br><br>The mapper always creates only one "true" way out of an intersection |
| `bearings` | Array<Integer> | `paths[i].instructions[i].<interval>`<br><br>`paths[i].points.<coordinates>` | Array<Integer><br><br>Array<Double> | *calculateBearing()*<br><br>returns the angle between this and the next intersection | |
| `location` | Array<Double> | `paths[i].instructions[i].<interval>`<br><br>`paths[i].points.<coordinates>` | Array<Integer><br><br>Array<Double> | no conversion needed: lookup index of interval in `<coordinates>` | |

**Voice Instruction Object**

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| `distanceAlongGeometry` | Integer | | Integer | set arbitrarily and subject to change | Distance to next maneuver at which the voiceInstruction should be announced. This is either FAR=2000m, ... , VERY_CLOSE = 200m |
| `announcement` | String | `paths[i].instructions[i].<text>` | String | | Can also include the next instruction's text, e.g. if the next instruction is a really short turn |
| `ssmlAnnouncement` | String | `paths[i].instructions[i].<text>` | String | *getSsmlAnnouncement()*<br><br>returns the announcement in ssml syntax | Mapbox Navigation will use the ssml announcement if valid |

**Banner Instruction Object**

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| distanceAlongGeometry | Integer | paths[i].instructions[i].<distance> | Integer | set arbitrarily and subject to change | Distance to next maneuver at which the voiceInstruction should be announced. This is either FAR=2000m, ... , VERY_CLOSE = 200m |
| primary | | | | | |
| primary.text | String | paths[i].instructions[i+1].<text> | String | No conversion needed | The text displayed is the text of the next! instruction, not the current one |
| primary.type | String | paths[i].instructions[i+1].<sign> | Integer | *getType()*<br><br>(See Maneuver Object type) | Maneuver type of next instruction |
| primary.modifier | String | paths[i].instructions[i+1].<sign> | Integer | *getMapboxModifier()*<br><br>(See Maneuver Object modifier) | Maneuver modifier of next instruction |
| components.text | String | (See primary.text) | | | |
| components.type | String | | | **text** | |

**RouteOptions Object**

| Mapbox | needed type | Used Graphhopper Data | type | conversion applied | Comment |
|---|---|---|---|---|---|
| baseUrl | String | | | **https://api.mapbox.com** | |
| user | String | | | **mapbox** | |
| profile | String | Request parameter: locale | | *convertProfile()*<br><br>(See Step Object mode) | |
| coordinates | Array<Double> | paths[i].snapped_waypoints.coordinates | Array<Double> | No conversion needed | |
| language | String | Request parameter: locale | | | |
| bearings | String | | | **";"** | |
| continueStraight | Boolean | | | **true** | |
| roundaboutExits | Boolean | | | **true** | |
| geometries | String | | | **polyline6** | |
| overview | String | | | **full** | |
| steps | Boolean | | | **true** | |
| annotations | String | | | **""** | |
| voiceInstructions | Boolean | | | **true** | |
| bannerInstructions | Boolean | | | **true** | |
| voiceUnits | String | Request parameter: locale | String | *getUnitSystem()*<br><br>returns either "metric" or "imperial" based on given language e.g. locale =en-us will return "imperial" | |
| accessToken | String | Request parameter: mapboxkey | String | no conversion needed | Mapboxkey can be passed along the request |
| requestUuid | String | ***No such data in GH response, has to generated while mapping*** | String | *generateUuid()* | |

Ignored Data from the GH Response:

- paths[i]
  - bbox
  - details