Optimization 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)

Mapbox Optimization Response Object

Mapbox	needed type	Used Graphhopper Data	type	Conversion	Comment
`code`	String			**Ok**	
`waypoints`	Array<WaypointObject>				
`trips`	Array<TripObject>				

Waypoint Object

Mapbox	needed type	Used Graphhopper Data	type	Conversion	Comment
`location`	Array<Double>	`solution.routes[i].activities[i].address.` `<lat>(<lon>)`	Double	*getLocation()* returns the Longitude and Latitude in an Array	
`waypoint_index`	Integer	`solution.routes[i].activities`	JSONArray	Index of activity used in current waypoint	

`name`	String	`solution.routes[i].activities[i].<name>`
		
		`solution.routes[i].activities[i].`
		`<location_id>`
			String	*getActivityName()*
				
				Name of current activity or location_id if name is not defined	

`trips_index`	Integer	`solution.routes`	JSONArray	Index of current route	

Trip Object

Mapbox	needed type	Used Graphhopper Data	type	Conversion	Comment
`distance`	Integer	`solution.` `<distance>`	Integer	no conversion needed: total distance of trip	
`duration`	Integer	`solution.` `<transport_time>`	Integer	no conversion needed: total duration of trip	

`geometry`	String (polyline)	`(See WaypointObject location)`
		
		`solution.routes[i].<points>`
				getGeometry()
				
				returns polyline6 encoded String of Array of all activity locations or respectively the "points" Array in routes.., if detailed response
					if calc_points is set to true in configuration object in the problem.json, an array of all visited points along all roads is included in the response. Can be extremely big.

`legs`	Array<Leg Object>				

Leg Object

Mapbox	needed type	Used Graphhopper Data	type	Conversion	Comment
summary	String	(See `WaypointObject  name`)	String	*getSummary()* returns "A to B" (A = current activity name, B= next activity name)	
steps	JSONArray	**no way to request turn-by-turn instructions for GH activities**		**[ ]**	Only GH Routing API can return turn-by-turn instructions
distance	Integer	`solution.routes[i].activities [i].distance`	Integer	*getActivityDistance()* returns accumulated distance of previous activity - acc. distance of current activity	waiting_time at an activity is possible in GH and is included in the conversion
duration	integer	`solution.routes[i].activities [i].<arr_time>/<end_time>`	Integer	*getActivityDuration()* returns end_time of previous activity - arrival_time of current activity	activity times are timestamps in GH

Ignored Data from the GH Response:

- `job_id`
- `status`
- `waiting_time_in_queue`
- `processing_time`
- `unassigned`
- `solution`
  - `costs`
  - `completion_time`
  - `max_operation_time`
  - `waiting_time`
  - `no_vehicles`
  - `no_unassigned`
  - `routes[i]`
    - `activities[i]`
      - `type`
      - `driving_time`
      - `load_before`