|  |  |  |
| --- | --- | --- |
| Job | | |
| Attribute | Description | Data example |
| Id | Integer id |  |
| Location | Coordiantes array |  |
| Location index | Index of relevant row and column in custom matrix |  |
| Service | Duration to do this task |  |
| Delivery | Array integers |  |
| Pickup | Array of integer pick up |  |
| Priority | Range 0 – 10 |  |
| Time\_windows | Time start, end (seconds) |  |

If a custom matrix is provided:

* location\_index is mandatory
* location is optional but can be set to retrieve coordinates in the response

If no custom matrix is provided:

* a table query will be sent to the routing engine
* location is mandatory
* location\_index is irrelevant

|  |  |  |
| --- | --- | --- |
| Vehicle | | |
| Attribute | Description | Data example |
| Id |  |  |
| Profile |  |  |
| Start | Coor array |  |
| Start\_index | Index of relevant row and column in custom matrix |  |
| End | Coordinates array |  |
| End\_index | Index of relevant row and column in custom matrix |  |
| Capacity | An array of integer describe multi-d quantities |  |
| Skills | Array of integer define skill |  |
| Time\_window | Working hour [start,end] |  |

**vehicle**

**locations**

* key start and end are optional for a vehicle, as long as at least one of them is present
* if end is omitted, the resulting route will stop at the last visited job, whose choice is determined by the optimization process
* if start is omitted, the resulting route will start at the first visited job, whose choice is determined by the optimization process
* to request a round trip, just specify both start and end with the same coordinates
* depending on if a custom matrix is provided, required fields follow the same logic than for job keys location and location\_index

### Capacity restrictions

Use amounts (capacity for vehicles, delivery and pickup for jobs) to describe a problem with capacity restrictions. Those arrays can be used to model custom restrictions for several metrics at once, e.g. number of items, weight, volume etc. A vehicle is only allowed to serve a set of jobs if the resulting load at each route step is lower than the matching value in capacity for each metric. When using multiple components for amounts, it is recommended to put the most important/limiting metrics first.

It is assumed that all delivery-related amounts for jobs are loaded at vehicle start, while all pickup-related amounts for jobs are brought back at vehicle end.

### Job priorities

Useful in situations where not all jobs can be performed, to gain some control on which jobs are unassigned. Setting a high priority value for some jobs will tend as much as possible to have them included in the solution over lower-priority jobs.

### Time windows

It is up to users to decide how to describe time windows:

* **relative values**, e.g. [0, 14400] for a 4 hours time window starting at the beginning of the planning horizon. In that case all times reported in output with the arrival key are relative to the start of the planning horizon;
* **absolute values**, "real" timestamps. In that case all times reported in output with the arrival key can be interpreted as timestamps.

The absence of a time window in input means no timing constraint applies. In particular, a vehicle with no time\_window key will be able to serve any number of jobs, and a job with no time\_windows key might be included at any time in any route, to the extent permitted by other constraints such as skills, capacity and other vehicles/jobs time windows.

## Matrix

A matrix object is an array of arrays of unsigned integers describing the rows of a custom travel-time matrix as an alternative to the travel-time matrix computed by the routing engine. Therefore, if a custom matrix is provided, the location, start and end properties become optional. Instead of the coordinates, row and column indications provided with the \*\_index keys are used during optimization.

## Summary

The summary object has the following properties:

| **Key** | **Description** |
| --- | --- |
| cost | total cost for all routes |
| unassigned | number of jobs that could not be served |
| service | total service time for all routes |
| duration | total travel time for all routes |
| waiting\_time | total waiting time for all routes |
|  |  |
| [delivery] | total delivery for all routes |
| [pickup] | total pickup for all routes |
| [distance]\* | total distance for all routes |

\*: provided when using the -g flag.

## Routes

A route object has the following properties:

| **Key** | **Description** |
| --- | --- |
| vehicle | id of the vehicle assigned to this route |
| [steps](https://github.com/VROOM-Project/vroom/blob/master/docs/API.md#steps) | array of step objects |
| cost | cost for this route |
| service | total service time for this route |
| duration | total travel time for this route |
| waiting\_time | total waiting time for this route |
|  |  |
| [delivery] | total delivery for jobs in this route |
| [pickup] | total pickup for jobs in this route |
| [geometry]\* | polyline encoded route geometry |
| [distance]\* | total route distance |

\*: provided when using the -g flag.

### Steps

|  |  |
| --- | --- |
| type | a string that is either start, job or end |
| arrival | estimated time of arrival at this step |
| duration | cumulated travel time upon arrival at this step |
| [location] | coordinates array for this step (if provided in input) |
| [job] | id of the job performed at this step, only provided if type value is job |
| [load] | vehicle load after step completion (with capacity constraints) |
| [service] | service time at this step, only provided if type value is job |
| [waiting\_time] | waiting time upon arrival at this step, only provided if type value is job |
| [distance]\* | traveled distance upon arrival at this step |