Table of Contents

1 Document Revision History

Date	Description
2014-07-22	Added support for time-variant traffic regions. See Table 7 and Table 8 for details.
2014-05-28	Added support for excluding specific vehicles in a fleet when creating a new Routeplan. See property *subfleetExcludeDrivers* at Chapter 4.3
2014-03-05	Added support for "Pickup and Delivery" dependency constraints. For more details see new property *pairedDeliveryIdx* in Table 4.
2013-04-18	Introduced *workingDayStartMin* parameter. Added Chapters 2.5, A.7 and A.8. Updated field descriptions in Table 3 and Table 5.
2013-02-04	Viamente Route Planner API v2.0, initial release

2 Introduction

Viamente Route Planner (VRP) exposes its data via an Application Programming Interface (API). This document is the official reference for that functionality.

2.1 Basic concepts

2.1.1 The API is entirely HTTP-based

Methods to retrieve data from the API require a GET request. Methods that submit, change, or destroy data require a POST. A DELETE request is also accepted for methods that destroy data. API Methods that require a particular HTTP method will return an error if you do not make your request with the correct one. HTTP Response Codes are meaningful.

2.1.2 The API is a RESTful resource

The API attempts to conform to the design principles of Representational State Transfer (REST). More details on REST can be found here: http://en.wikipedia.org/wiki/Representational_state_transfer
Libraries to build REST-compatible API clients are readily available for several programming languages. A list of REST frameworks is available at the bottom of the above mentioned page.

2.1.3 The format used for structured data exchange is JSON

The API supports the JSON (JavaScript Object Notation) format. Details on how JSON works can be found here: http://json.org/ and here: http://en.wikipedia.org/wiki/JSON
Libraries to convert to and from the JSON format are readily available for popular and less popular programming languages. A full index, sorted by language, can be found at the bottom of this page: http://json.org/

2.1.4 SSL is supported (and highly recommended)

The API works with both plain HTTP and HTTP over SSL (HTTPS).
SSL is highly recommended to avoid passing confidential data (e.g.: client's names and addresses) in clear text over the web.

2.1.5 Parameters have certain expectations

Some API methods take optional or requisite parameters. Please keep in mind that all query-string parameter values should be converted to UTF-8 and URL encoded.

2.1.6 Concurrency

There is a limit on the maximum number of API requests that can be concurrently submitted from the same account.

This limit is "1", that is: each API account can run one request at a time.

2.1.7 Rate limiting

API requests are rate limited. If the service receives too many requests from the same account within the allowed time period it will reject the exceeding requests returning an HTTP 429 code.

Some API methods, like POST requests to the "routeplans" resource, have specific rate limits which are applied in conjunction with the main REST API rate limit. Calls to such methods will count against both the method and default request rate limit. If either limit is exhausted, the method will fail with an HTTP 429 code.

Rate limiting is handled according to the "leaky bucket" model.

2.2 Authentication

Authentication is achieved by passing the API key in each API request.

For example:

```
https://vrp.viamente.com/api/vrp/v2/routeplans?key=YOUR_API_KEY
```

2.2.1 Obtaining an API key

To obtain an API key: sign in to your Viamente Route Planner Account at https://vrp.viamente.com then click on "Account" then on "API Integration".

From the API Integration panel **verify that your Subscription Plan qualifies for API access**.

If your Account qualifies, make sure "Enable API Integration" is checked and copy the "API Key" value.

2.3 REST API Resources

The Viamente Route Planner API includes the following RESTful resources to integrate with the Viamente Route Planner web application. Each resource is further detailed later in its own Chapter.

2.3.1 Subfleet handling Resource

```
https://vrp.viamente.com/api/vrp/v2/subfleets
```

2.3.2 Routeplan handling Resource

```
https://vrp.viamente.com/api/vrp/v2/routeplans
```

2.4 HTTP Response Codes and Errors

2.4.1 HTTP Status Codes

The Viamente API attempts to return appropriate HTTP status codes for every request.

- **200 OK**: Success!
- **400 Bad Request**: The request was invalid. An accompanying error message will explain why.
- **401 Unauthorized**: The `key` parameter is missing.
- **403 Forbidden**: Invalid API key.
- **404 Not Found**: The URI requested is invalid or the resource requested does not exists.
- **429 Too Many Requests**: The user has sent too many requests in a given amount of time.
- **500 Internal Server Error**: Something bad happened. You may try again later but please contact Viamente support team so that they may investigate.
- **502 Bad Gateway**: The API service is temporarily down or undergoing maintenance.
- **503 Service Unavailable**: The Viamente servers are up, but overloaded with requests. Try again later.

2.4.2 Error Messages

When the Viamente API returns error messages (HTTP status code != 200), it does so in JSON format. For example, an error might look like this:

```
{res: "ERR","errorCode": -1000,"errorDescription": "Failed parsing JSON
source"}
```

2.4.3 Error Codes

Error codes, as found in the body of returned error messages, further define the scope of an error. The following error codes may be returned:

- **-100: Internal Server Error**: Something bad happened. You may try again later but please contact Viamente support team so that they may investigate.
- **-900: Too Many Requests**: The given key has submitted too many requests in too short a period of time. Insert a pause between requests to avoid getting this error.
- **-901: Too Many Concurrent Requests**: The given key has submitted too many concurrent requests. Wait for previously submitted requests to complete to avoid getting this error.
- **-902: Key Daily Limit Exceeded**: The maximum daily amount of requests has been reached. Further requests must wait till the next day. More information is provided in the `errorDescription` field.
- **-1000: Malformed Request**: something is wrong with the input. Either a query-string parameter is missing or malformed or the JSON data in the request body is malformed (typically due to a missing or overabundant parenthesis or comma or a missing mandatory field).

2.5 Working during the night

In the Viamente Route Planner a "working day" lasts 24 hours and, by default, starts at 0:00 (therefore ending at 24:00 and matching the "solar" day).

Since a "working day" lasts 24 hours, a Route starting in day 1 must be completed before midnight of the same day and cannot go over midnight and "overflow" into day 2.

This is ok when all work happens during the day, say between 7am and 6pm, but it does not fit a scenario where Orders are scheduled during the night so that Drivers have to work across midnight, say between 8pm and 6am.

Such "across midnight" scenarios can be handled by redefining when the 24-hours "working day" starts.

In the above mentioned scenario, for example, the "working day" could be set to start at 1pm and, consequently, end 24 hours later at 1pm the next "solar" day. This would allow Drivers to start working at 8pm and continue driving after midnight (and up to the end of the working day).

3 Subfleet Resource

A Subfleet is a group of Drivers saved together with their individual settings (start and destination address, working time window, load capacity, working days, etc.).

Note: Subfleets can only be created online after logging in to the Viamente Route Planner web application.

3.1 List Subfleets

Description:

List the available Subfleets.

URL:

```
https://vrp.viamente.com/api/vrp/v2/subfleets
```

Formats:

json

HTTP Method(s):

GET

Return values:

Property	Type	Description
subfleets	Array of Subfleet	Array of Subfleet objects.

Subfleet-details object properties:

Property	Type	Description
id	String	System-assigned unique fleet ID.
name	String	User-assigned subfleet name.
drivers	Integer	Subfleet size in terms of number of Drivers.
createDate	Date	Subfleet creation UTC date formatted as "yyyy-MM-dd HH:mm:ss".

Response body example:

```
{
  "subfleets": [{
    "id": "flt-eed9604d-f870-4229-9758-a63de01fde8e",
    "name": "Saturday subfleet",
    "drivers": 5,
    "createDate": "2011-05-26 13:25:05"
  }, {
    "id": "flt-e283086b-58b3-468e-98a0-712538b0e5ae",
    "name": "Weekly subfleet",
    "drivers": 4,
    "createDate": "2011-07-12 19:49:25"
  }, {
    "id": "flt-921073af-2f00-459d-9cad-b3b0d3d5f70c",
    "name": "Upper east side",
    "drivers": 13,
    "createDate": "2012-03-06 16:49:56"
  }, {
    "id": "flt-78c43be9-a522-4e3e-af7d-eb8ebbdca199",
    "name": "Tom\u0027s team",
    "drivers": 3,
    "createDate": "2012-05-10 19:43:43"
  }]
}
```

3.2 Get a Subfleet

Description:

List Drivers that are part of a given Subfleet.

URL:

```
https://vrp.viamente.com/api/vrp/v2/subfleets/{subfleetID}
```

Formats:

json

HTTP Method(s):

GET

Return values:

Property	Type	Description
id	String	System-assigned unique subfleet ID.
name	String	User-assigned subfleet name.
createDate	Date	Subfleet creation UTC date formatted as "yyyy-MM-dd HH:mm:ss".
drivers	Array of Driver	Array of Driver objects.

Table 1: Driver object properties

Property	Type	Description
id*	Integer	Unique Driver identifier.
name	String	User-assigned Driver name.
email	String	Driver email.
dayToSettings*	Map	This is a key/value map (or associative array) containing one entry for each defined working day for this Driver. The key is the working day number (1,2,…,N) and the associated value is a DriverSettings object defining settings applied to that specific working day only. Working days start from 1 and are limited according to the maximum number of working days (aka "planning horizon") allowed by the purchased Subscription Plan.

Table 2: DriverSettings object properties. **All properties apply to a specific working day only**.

Property	Type	Description
maxCapacity	Integer	Maximum vehicle capacity in load units #1 (e.g.: volume in cm3).
maxCapacity2	Integer	Maximum vehicle capacity in load units #2 (e.g.: weight in kg).
maxCapacity3	Integer	Maximum vehicle capacity in load units #3 (e.g.: goods value is $).
timeWindow*	TimeWindow	Defines the Driver availability time range. The Driver's time window will take into account the service time at the last visited Order and, if a destination is specified, the driving time to

		get there. See Table 5 for TimeWindow object properties.
maxWorkingTimeMin	Integer	Maximum allowed working time from departure to arrival at the end destination. "Working time" is the sum of driving time + service time at each served Order + idle time (if any) + break time (if any). If not set or set to 0, the maximum working time is equal to the Driver's availability timeWindow.
maxDrivingTimeMin	Integer	Maximum allowed driving time from departure to arrival at the end destination. If not set or set to 0, the maximum driving time is equal to the Driver's availability timeWindow.
fixedStartTime	Boolean	When set to false, the optimization engine chooses the departure time (within the given time window) that minimizes the overall working time. If set to true the optimization engine forces the Driver to depart at the beginning of the working time window. This leads to routes that, on average, are completed earlier but with a potentially longer working time. For example: given a working Time Window between 8am and 6pm, with fixedStartTime set to false the output schedule could be 9am-5pm. With fixedStartTime set to true the output schedule could be 8am-4.30pm. **Note:** this parameter is only relevant when, in a Routeplan, one or more Orders have time window constraints. **Note:** this parameter maps to the "flexible start time" checkbox in the Edit Driver UI dialog.
activationCostCent	Integer	A setup cost (in, for example, USD cents) that must be taken into account only if this Driver is required to work. If a RoutePlan includes multiple Drivers and not all Drivers are necessary to serve all Orders, Drivers with higher activationCost are less likely to be used.
hourlyCostCent	Integer	The cost for one hour of driving time (e.g.: the Driver's hourly wage and/or the hourly rental cost for the vehicle). If a RoutePlan includes multiple Drivers and not all Drivers are necessary to serve all Orders, Drivers with higher hourly cost are less likely to be used, and if they must be used they are assigned to shorter routes.
includeRegion	Array of Location	A list of 3 or more Locations specifying the shell of a simple closed polygon (which can be concave, convex or self-intersecting). If specified, only this Driver will serve Orders falling inside the defined region (including edges), while still complying with all other constraints.
origin*	Location	The location of the starting point for this Driver. See Table 6 for Location object properties.
destination	Location	If provided, defines the location of the end-destination for this Driver. If not specified, the route calculated for this Driver ends at the last visited Order. See Table 6 for Location object properties.
skills	Array of String	List of textual skills associated to this Driver. Used together with Order's **skillExclude** and **skillsInclude** fields, skills enable configuring complex Driver-to-Order association constraints.
break	Break	If provided, defines break constraints for this Driver.

		See Table 3 below for Break object properties.
speedPercent	Integer	This is a percentage value between 50 and 200 and is a modifier affecting driving times for this specific vehicle. Set below 100 if this vehicles travels at a lower-than-average speed, or above 100 if it travels at a higher-than-average speed. See also the **driversSpeedPercent** Routeplan setting for a global driving times modifier affecting all vehicles. Defaults to 100.
notes	String	Planner notes.

Table 3: Break object properties

Property	Type	Description
startMin*	Integer	Minimum allowed break start time in minutes since midnight (00:00). The allowed range spans a maximum of 24 hours [0 – 1440].
stopMin*	Integer	Maximum allowed break stop time in minutes since midnight (00:00). The allowed range can span a maximum of 24 hours [0 – 1440]. If stopMin is smaller than startMin, than it is considered to be after midnight (after 1440 minutes). Example 1: startMin = 1080 (18:00), stopMin = 300 (5:00): stopMin is considered to be 5:00 *after* midnight.
durationMin*	Integer	Break duration in minutes The break can be scheduled to start at any time between startMin and stopMin, and lasts for durationMin. Example: startMin = 780 (13:00), stopMin = 810 (13:30) and durationMin = 60 (1 hour) means: schedule a 1-hour break starting at any time between 13:00 and 13:30.

Response body example:

```
{
  "id": "flt-78c43be9-a522-4e3e-af7d-eb8ebbdca199",
  "name": "Tom \u0027s team",
  "createDate": "2012-05-10 19:43:43",
  "drivers": [{
    "id": 1,
    "name": "Alice",
    "required": false,
    "dayToSettings": {
      "1": {
        "maxCapacity": 120,
        "maxCapacity2": 0,
        "maxCapacity3": 0,
        "timeWindow": {
          "startMin": 480,
          "stopMin": 1020
        },
        "maxWorkingTimeMin": 0,
        "maxDrivingTimeMin": 0,
        "fixedStartTime": true,
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "includeRegion": [{
          "lat": 40.745176,
          "lng": -74.223632
        }, {
          "lat": 40.699901,
          "lng": -74.230499
        }, {
          "lat": 40.712914,
          "lng": -74.148101
        }],
        "origin": {
          "address": "Jelliff Ave, Newark",
          "lat": 40.72663,
          "lng": -74.19483
        },
        "skills": [
          "plumbing",
          "high-capacity"],
        "break": {
          "startMin": 720,
          "stopMin": 750,
          "durationMin": 60
        },
        "notes": ""
      },
      "5": {
        "maxCapacity": 120,
        "maxCapacity2": 0,
        "maxCapacity3": 0,
        "timeWindow": {
          "startMin": 480,
          "stopMin": 1020
        },
        "maxWorkingTimeMin": 0,
        "maxDrivingTimeMin": 0,
        "fixedStartTime": true,
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "includeRegion": [],
        "origin": {
          "address": "Jelliff Ave, Newark",
          "lat": 40.72663,
          "lng": -74.19483
        },
        "skills": [
          "plumbing",
          "high-capacity"],
        "break": {
          "startMin": 720,
          "stopMin": 750,
          "durationMin": 60
        },
        "notes": ""
      }
    }
  }, {
    "id": 2,
    "name": "Bob",
    "required": false,
    "dayToSettings": {
      "1": {
        "maxCapacity": 80,
        "maxCapacity2": 0,
        "maxCapacity3": 0,
        "timeWindow": {
          "startMin": 480,
          "stopMin": 1020
        },
        "maxWorkingTimeMin": 0,
        "maxDrivingTimeMin": 0,
        "fixedStartTime": true,
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "includeRegion": [],
        "origin": {
```

```
          "address": "380 Washington St, Newark",
          "lat": 40.73186,
          "lng": -74.17754
        },
        "destination": {
          "address": "380 Washington St, Newark",
          "lat": 40.73186,
          "lng": -74.17754
        },
        "skills": [
          "roof-repair"],
        "break": {
          "startMin": 780,
          "stopMin": 840,
          "durationMin": 60
        },
        "notes": ""
      }
    }
  }, {
    "id": 3,
    "name": "Carl",
    "required": false,
    "dayToSettings": {
      "1": {
        "maxCapacity": 100,
        "maxCapacity2": 0,
        "maxCapacity3": 0,
        "timeWindow": {
          "startMin": 480,
          "stopMin": 1020
        },
        "maxWorkingTimeMin": 0,
        "maxDrivingTimeMin": 0,
        "fixedStartTime": true,
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "includeRegion": [],
        "origin": {
          "address": "30 Hamilton St, Newark",
          "lat": 40.73259,
          "lng": -74.16825
        },
        "skills": [
          "soldering"],
        "break": {
          "startMin": 750,
          "stopMin": 810,
          "durationMin": 60
        }
      }
    }
  }]
}
```

4 Routeplan Resource

A Routeplan is a collection of Orders that need to be served and of Drivers that are supposed to serve them. Once uploaded via this API, a Routeplan can be opened from the Viamente Route Planner where routes can be built automatically using the "Plan Optimal Routes" button and then, optionally, manually adjusted.
The resulting optimized routes can then be exported using this API.

Note: A Routeplan is typically *not* a collection of all possible Orders (e.g.: whole list of customers), but of a specific subset of Orders that need to be served on a specific occasion (e.g.: "Monday's Deliveries" or "New York's Orders" or "3rd Week Deliveries"). Similarly, not all Drivers necessarily need to be added to a Routeplan, but just that subset of Drivers (subfleet) required to serve the Routeplan Orders.

4.1 List Routeplans

Description:

List saved Routeplans.

URL:

`https://vrp.viamente.com/api/vrp/v2/routeplans`

Formats:

json

HTTP Method(s):

GET

Return values:

Property	Type	Description
routeplans	Array	Array of Fleet objects.

Routeplan-details object properties:

Property	Type	Description
id	String	System-assigned unique Routeplan ID.
name	String	User-assigned Routeplan name.
label	String	User-assigned Routeplan label text.
drivers	Integer	Routeplan size in terms of number of Drivers.
orders	Integer	Routeplan size in terms of number of Orders.
horizonDays	Integer	Routeplan horizon in terms of number of days.
createDate	Date	Routeplan creation UTC date formatted as "yyyy-MM-dd HH:mm:ss".
lastUpdateDate	Date	Routeplan last update UTC date formatted as "yyyy-MM-dd HH:mm:ss".

Response body example:

```
{
  "routeplans": [{
    "id": "rpl-fa0f821e-5334-4e5f-b0d8-ff128d20b71a",
    "name": "New York deliveries, 7 May 2012",
    "label": "Some notes or a useful label text here",
    "drivers": 10,
    "orders": 232,
    "horizonDays": 1,
    "createDate": "2012-05-06 15:57:36",
    "lastUpdateDate": "2012-05-07 11:32:15"
  }, {
    "id": "rpl-1d889dd4-0598-4114-a7a4-0b6eee3bf63e",
    "name": "New York deliveries, 14 May 2012",
    "label": "NY",
    "drivers": 10,
    "orders": 241,
    "horizonDays": 1,
    "createDate": "2012-05-13 16:05:48",
    "lastUpdateDate": "2012-05-14 11:02:54"
  }, {
    "id": "rpl-742cfd36-9970-4196-9c04-bea7dc3253c7",
    "name": "Tom\u0027s routes",
    "label": "NJ",
    "drivers": 3,
    "orders": 45,
    "createDate": "2012-06-01 16:06:37",
    "lastUpdateDate": "2012-06-01 18:18:29"
  }, {
    "id": "rpl-6cf4cb13-5d21-448e-9b65-2fcace57098a",
    "name": "3rd Week Deliveries",
    "label": "NJ",
    "drivers": 3,
    "orders": 246,
    "horizonDays": 5,
    "createDate": "2013-01-23 12:16:21",
    "lastUpdateDate": "2013-01-24 19:10:55"
  }]
}
```

4.2 Get a Routeplan

Description:

Show Routeplan data.

URL:

```
https://vrp.viamente.com/api/vrp/v2/routeplans/{planID}
```

Formats:

json

HTTP Method(s):

GET

Return values:

Property	Type	Description
id	String	System-assigned unique Routeplan ID.
name	String	User-assigned Routeplan name.
label	String	User-assigned Routeplan label text.
createDate	Date	Routeplan creation UTC date formatted as "yyyy-MM-dd HH:mm:ss".
lastUpdateDate	Date	Routeplan last update UTC date formatted as "yyyy-MM-dd HH:mm:ss".
horizonDays	Integer	The planning horizon, in number of days, set for this Routeplan.
workingDayStartMin	Integer	Marks the beginning, in minutes since midnight, of the 24-hours working day for this Routeplan. See also Chapter 2.5.
driversSpeedPercent	Integer	This is a percentage value between 50 and 200 and is a modifier affecting all driving times. Set to a lower value if your vehicles travel at a lower-than-average speed, or set to a higher value if they travel at a higher-than-average speed. **Note**: driversSpeedPercent can also be used to model global traffic conditions. For local traffic conditions use trafficProfile.
trafficProfile	TrafficProfile	The traffic profile applied to this Routeplan (if any). A traffic profile is defined by multiple regions, each one with its own traffic level. **Note:** driversSpeedPercent and trafficProfile are independently applied. See Table 7 for TrafficProfile object details.
loadUnitType	String	Define what a load unit #1 is in this Routeplan. This does not affect the optimization process and is simply a string used when exporting route data to provide a name for the load units. Examples for this value are: pounds, gallons, cubic meters, passengers, crates, dollars, goods, etc.
loadUnit2Type	String	Define what a load unit #2 is in this Routeplan. The same considerations enunciated for loadUnitType apply here.
loadUnit3Type	String	Define what a load unit #3 is in this Routeplan. The same considerations enunciated for loadUnitType apply here.
drivers	Array of Driver	List of all Drivers included in this Routeplan. See Table 1 at Paragraph 3.2 for Driver object details.
orders	Array of Order	List of all Orders included in this Routeplan. See Table 4 below for Order object details.

routes	Array of Route	List of all optimized Routes included in this Routeplan. See Table 9 below for Route object details. **Note:** this list will be empty for a newly created Routeplan. Routes need to be built from the Viamente Route Planner (and the updated Routeplan needs to be saved) before they can be retrieved with this API.

Table 4: Order object properties

Property	Type	Description	Def. Value
name	String	If provided, this name string can be used to identify this waypoint in the output.	-
notes	String	Generic notes field associated with this Order.	-
location*	Location	Defines the Order's location via a street address string and its geocoded latitude/longitude coordinates.	-
loadUnits	Integer	Defines the amount of the delivery in terms of load units #1 (i.e.: consumed vehicle capacity). A positive value indicate a delivery, a negative one indicates a pickup. **Note:** since each Order can only be either a pickup or a delivery (not both at the same time), when multiple loadUnits are provided (loadUnits, loadUnits2, loadUnits3) they have to either be all >= 0 (delivery) or all <=0 (pickup). Providing loadUnits with mixed signs results in a warning and in all load values being ignored (set to 0).	0
loadUnits2	Integer	Defines the amount of the delivery in terms of load units #2. Limits and considerations described at loadUnits also apply.	0
loadUnits3	Integer	Defines the amount of the delivery in terms of load units #3. Limits and considerations described at loadUnits also apply.	0
pairedDeliveryIdx	Integer	Used to define a "Pickup & Delivery" constraint. Defines the 0-based index of the paired delivery Order in the orders array. **Note:** when pairedDeliveryIdx is set: • this Order will be considered a "pickup" (the sign of the loadUnits values is ignored) • the Order referenced by pairedDeliveryIdx will be considered a "delivery" and its loadUnits will be ignored and replaced by those defined by this Order (all objects that are loaded at the pickup location are considered to be a single indivisible "package" that must be unloaded as-is at the delivery location). **Note:** a given delivery Order cannot be referenced by more than one pickup Order. Subsequent duplicated references result in a warning and are ignored. If goods loaded from several different pickup locations must all be unloaded at the same delivery location then multiple pickup/delivery Order pairs must be created (one for each indivisible "package") to model that scenario. The same is true if multiple goods picked up at one location must be delivered to multiple different locations.	-

		Note: a delivery Order already referenced by a pickup Order cannot itself define pairedDeliveryIdx. If this happens a warning is issued and pairedDeliveryIdx is ignored. If a pickup must take place at the same location of a delivery then an additional pickup/delivery Order pair must be created to model that scenario.	
serviceTimeMin	Integer	The amount of time, in minutes, that the Driver is expected to spend servicing this Order (this may include, for example, estimated parking time, time to get from the parked vehicle to the floor, estimated service time, etc.).	0
timeWindows	Array of TimeWindow	Defines an array of up to 2 arrival time window constraints for this Order. Each time window defines a time interval when the Order can start to be serviced. This interval does not include the service time (e.g.: if service time is 30 minutes and the time window is from 10:00 to 11:00, then it is ok for the Driver to arrive and start servicing the Order at 10:59 and complete the service 30 minutes later at 11.29). If the time window must include the service time, just subtract the service time from the end time of the time window (e.g.: if service time is 30 minutes and the time window is from 10:00 to 11:00 and must include the service time, then change the time window to be from 10:00 to 10:30). Up to two time windows can be used, for example, to prevent the Order from being serviced in off-limits intervals (e.g.: defining the time windows: 8:00-12:00 and 14:00-18:00 would prevent the Driver from arriving at the Order between 12:00 and 14:00). If not specified, this Order will always be available to be serviced. Limitations: • the duration of a single time window cannot exceed 24 hours • all defined time windows must fit into the 24 hours span defined between workingDayStartMin and workingDayStartMin + 24 hours See table below for TimeWindow object properties.	(no constraint)
eligibleDays	Array of Integers	Defines on which days this Order can be served. Valid values range from 1 to horizonDays (inclusive). An empty array means "any day". Invalid/inconsistent values are ignored.	(no constraint)
importance	String	Defines the importance of this Order. Can be any one of: lowest, lower, normal, higher, highest. **Note:** Importance is only taken into account when all Orders cannot be serviced by the given Drivers: under such circumstances Orders with a higher importance are preferred over lower-importance ones.	0
skillsInclude	Array of String	Define skill-based Driver inclusion constraints: only Drivers defining *all skills in this list* (AND-criteria) are allowed to service this Order.	-

skillsExclude	Array of String	Define skill-based Driver exclusion constraints: all Drivers defining *one or more skills in this list* are not allowed to service this Order. Exclusion constraints take precedence over inclusion ones.	-
forceDriverID	Integer	If defined forces this Order to be served by the given Driver. **Note:** if defined, forceDriverID overrides both skillsInclude and skillsExclude.	-
customFields	Map	This is a key/value map (or associative array) defining custom text fields that will be added to the exported routes. The map key is the custom field name, the map value is the text associated to that field name. Each Order cannot define more than 10 custom fields. All Orders included in a single Routeplan cannot define more than 10 different custom field names.	-

Table 5: TimeWindow object properties

Property	Type	Description	Def. Value
startMin*	Integer	Time window start time in minutes since midnight (00:00). The allowed range can span a maximum of 24 hours [0 – 1440]. Values exceeding 1440 are accepted but implicitly modulo-ed by 1440 (e.g.: 1740 % 1440 -> 300). Set to -1 for "start of working day", where "start of working day" is defined as workingDayStartMin in the Routeplan object. **Note:** what follows is only relevant in scenarios where Drivers work during the night. See also Chapter 2.5. Where >= 0, startMin is interpreted based on workingDayStartMin in the Routeplan object. If startMin < workingDayStartMin then startMin is assumed to be after midnight. Example: if workingDayStartMin is set to 1080 minutes (18:00) and startMin is 300 minutes (5:00), then startMin is interpreted as 1440+300 minutes (5:00 of the following "solar" day).	-1
stopMin*	Integer	Time window stop time in minutes since midnight (00:00). The allowed range can span a maximum of 24 hours [0 – 1440]. Values exceeding 1440 are accepted but implicitly modulo-ed by 1440 (e.g.: 1740 % 1440 -> 300). Set to -1 for "end of working day", where "end of working day" is defined based on workingDayStartMin in the Routeplan object (workingDayStartMin + 24 hours). **Note:** what follows is only relevant in scenarios where Drivers work during the night. See also Chapter 2.5. If stopMin is smaller than startMin, than it is considered to be after midnight (after 1440 minutes). Example: startMin = 1080 (18:00), stopMin = 300 (5:00): stopMin is	-1

		considered to be 5:00 of the following "solar" day (1440+300=1740 minutes).	

Table 6: Location object properties

Property	Type	Description	Def. Value
address*	String	Full address like: 4514 Foster Avenue, Brooklyn, New York, NY 11203, United States Partial addresses (e.g.: without postcode) are accepted as well, but the risk of an inaccurate geocoding is higher.	
lat*	Double	Latitude coordinate specified using the WGS 84 reference frame. E.g.: 40.639545	-
lng*	Double	Longitude coordinate specified using the WGS 84 reference frame. E.g.: -73.933376	-
geoAddress	String	Returned value only. Do not pass when creating a new Routeplan. The normalized address, as returned by the geocoding/reverse geocoding process.	-
geoAccuracy	Integer	Returned value only. Do not pass when creating a new Routeplan. An indication of the accuracy of the geocoded address. Values are: 0,1,2,3: country/state/region accuracy 4: locality/sublocality accuracy 5: postal code accuracy 6: street accuracy 7: intersection accuracy 8: street number accuracy	-
geoStatus	Integer	Returned value only. Do not pass when creating a new Routeplan. Status of the geocoding process. Values are: 0: Geocoding/reverse geocoding successful 100: Unknown/Unrecognized address 200: Unsupported country 300: Low accuracy 400: Postcode mismatch 500: Postcode level accuracy	

IMPORTANT: when defining an Order's location the Location object can be populated:

1. Setting the *address* field only. In this case the provided address string is geocoded to find the latitude and longitude coordinates on the map.
   E.g.: address:"4514 Foster Avenue, Brooklyn, New York, NY 11203, United States" geocodes to latitude: 40.639545, longitude:-73.933376
2. Setting the *lat* and *lng* coordinates only. In this case the provided latitude and longitude coordinates are reverse-geocoded to populate the address field. The address field is only used for visualization.
   E.g.: latitude: 40.639545, longitude:-73.933376 reverse-geocode to "4514 Foster Avenue, Brooklyn, New York, NY 11203, United States"

3. Setting both. In this case the Order's position on the map is determined by the given coordinates and the string provided in the address field is only used for visualization.
   E.g.: latitude: 40.639545, longitude:-73.933376, address: "Tom's house"

Table 7: TrafficProfile object properties

Property	Type	Description	Def. Value
name*	String	A mnemonic name assigned to this Traffic Profile	-
description	String	A description for this Traffic Profile.	
timeWindows new	Array of TimeWindow	An array of up to two Time Windows defining non-overlapping intervals. Determines when time-depended traffic factors (TrafficRegion.twTrafficLevelPercent) are applied.	-
trafficRegions*	Array of TrafficRegion	Array of Traffic Regions that make up this Traffic Profile. See Table 8 below for details.	-

Table 8: TrafficRegion object properties

Property	Type	Description	Def. Value
trafficLevelPercent	Integer	This is a percentage value between 0 and 500 and is a modifier affecting driving times for arcs intersecting this region. Set according to the default expected traffic conditions for this region. 0% means no traffic, a 100% impact means that driving times will be twice as long, a 200% impact means that driving times will be three times longer, and so on.	0
twTrafficLevelPercent new	Array of Integer	Values overriding trafficLevelPercent when the arc is traveled during the corresponding Time Window (defined at TrafficProfile.timeWindows).	-
region*	Array of Location	A list of 3 or more Locations specifying the shell of a simple closed polygon. **Note:** Only the lat and lng Location fields are supported here. The address field is ignored. **Note:** Self-intersecting polygons are not directly supported (they are automatically converted to the enclosing convex polygon).	-

Table 9: Route object properties

Property	Type	Description	Def. Value
driverID	Integer	The ID of the Driver assigned to this route.	-
driverIdx	Integer	The zero-based index of the Driver in the returned drivers array.	-
day	Integer	The day of the route.	-
stops	Array of RouteStop	Ordered list of route stops. Route stops include one origin, one or more Orders, zero or one break, zero or one destination (route ends at the last visited Order if no destination is specified for the Driver).	

Table 10: RouteStop object properties

Type = "origin"

Property	Type	Description
type	String	"origin".
departureTimeSec	Integer	Departure time in seconds since midnight.
nextStopDriveTimeSec	Integer	Drive time to the next stop (excluding breaks), in seconds since midnight.

Type = "break"

Property	Type	Description
type	String	"break".
breakTimeSec	Integer	Break start time in seconds since midnight.
breakDurationSec	Integer	Break duration in seconds.

Type = "order"

Property	Type	Description
type	String	"order".
orderIdx	Integer	Zero-based index of the Order in the orders array.
arrivalTimeSec	Integer	Arrival time in seconds since midnight.
idleTimeSec	Integer	Time spent waiting for this Order's time window to open, in seconds. This is > 0 only if the scheduled arrival is earlier that the Order's time window opening time. **Note:** optimization criteria work to minimize the occurrence and the amount of idle time, but when time window constraints are involved this may not be completely avoidable.
serviceStartTimeSec	Integer	Service start time in seconds since midnight (= arrivalTimeSec + idleTimeSec).
departureTimeSec	Integer	Departure time in seconds since midnight (= serviceStartTimeSec + order service time)
nextStopDriveTimeSec	Integer	Drive time to the next stop (excluding breaks), in seconds since midnight.

Type = "destination"

Property	Type	Description
type	String	"destination".
arrivalTimeSec	Integer	Arrival time in seconds since midnight.

Response body example:

```
{
  "id": "rpl-3617e5b4-fc8d-4cf3-8a61-cf8f038ba37a",
  "name": "Lancaster deliveries",
  "label": "Some useful notes here",
  "createDate": "2013-02-04 15:56:47",
  "lastUpdateDate": "2013-02-04 18:10:47",
  "horizonDays": 2,
  "driversSpeedPercent": 100,
  "trafficProfile": {
    "name": "Lancaster",
    "description": "",
    "timeWindows": [{
      "startMin": 420,
      "stopMin": 540
    }, {
      "startMin": 960,
      "stopMin": 1140
    }],
    "trafficRegions": [{
      "trafficLevelPercent": 50,
      "twTrafficLevelPercent": [85, 120],
      "region": [{
        "lat": 34.70761,
        "lng": -118.17564
      }, {
        "lat": 34.708033,
        "lng": -118.23143
      }, {
        "lat": 34.661028,
        "lng": -118.243618
      }, {
        "lat": 34.626004,
        "lng": -118.216323
      }, {
        "lat": 34.631512,
        "lng": -118.1571
      }, {
        "lat": 34.693356,
        "lng": -118.15195
      }, {
        "lat": 34.70013,
        "lng": -118.16122
      }]
    }]
  },
  "loadUnitsType": "Goods",
  "loadUnits2Type": "Pounds",
  "loadUnits3Type": "Cubic Feet",
  "drivers": [{
    "id": 1,
    "name": "Alice",
    "required": false,
    "dayToSettings": {
      "1": {
        "notes": "",
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "skills": [],
        "fixedStartTime": false,
        "maxWorkingTimeMin": 0,
        "origin": {
          "address": "2739 W Avenue L, Lancaster, CA 93536, USA",
          "lat": 34.66031,
          "lng": -118.17887,
          "geoAddress": "2739 West Avenue L, Lancaster, CA 93536, USA",
          "geoAccuracy": 8
        },
        "timeWindow": {
          "startMin": 360,
          "stopMin": 600
        },
        "speedPercent": 100,
        "maxCapacity": 9,
        "maxCapacity2": 140,
        "maxCapacity3": 0,
        "includeRegion": [{
          "lat": 34.713395809196,
          "lng": -118.25958251953
        }, {
          "lat": 34.600150184591,
          "lng": -118.24378967285
        }, {
          "lat": 34.584888332278,
          "lng": -118.13461303711
        }, {
          "lat": 34.720168693111,
          "lng": -118.16825866699
        }]
      },
      "2": {
        "notes": "",
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "skills": [],
        "fixedStartTime": false,
        "maxWorkingTimeMin": 0,
        "origin": {
          "address": "2739 W Avenue L, Lancaster, CA 93536, USA",
          "lat": 34.66031,
          "lng": -118.17887,
          "geoAddress": "2739 West Avenue L, Lancaster, CA 93536, USA",
          "geoAccuracy": 8
        },
        "timeWindow": {
          "startMin": 360,
          "stopMin": 600
        },
        "speedPercent": 100,
        "maxCapacity": 9,
        "maxCapacity2": 140,
        "maxCapacity3": 0,
        "includeRegion": [{
          "lat": 34.713395809196,
          "lng": -118.25958251953
        }, {
          "lat": 34.600150184591,
          "lng": -118.24378967285
        }, {
          "lat": 34.584888332278,
          "lng": -118.13461303711
        }, {
          "lat": 34.720168693111,
          "lng": -118.16825866699
        }]
      }
    }
  }, {
    "id": 2,
    "name": "Bob",
    "required": false,
    "dayToSettings": {
      "1": {
        "notes": "",
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "skills": ["alice"],
        "fixedStartTime": false,
        "maxWorkingTimeMin": 0,
        "origin": {
          "address": "2739 W Avenue L, Lancaster, CA 93536, USA",
          "lat": 34.66031,
          "lng": -118.17887,
          "geoAddress": "2739 West Avenue L, Lancaster, CA 93536, USA",
          "geoAccuracy": 8
        },
        "destination": {
          "address": "2739 W Avenue L, Lancaster, CA 93536, USA",
          "lat": 34.66031,
          "lng": -118.17887,
          "geoAddress": "2739 West Avenue L, Lancaster, CA 93536, USA",
          "geoAccuracy": 8
        },
        "timeWindow": {
          "startMin": 600,
          "stopMin": 1140
        },
        "speedPercent": 100,
        "maxCapacity": 8,
        "maxCapacity2": 80,
        "maxCapacity3": 0
      },
      "2": {
        "notes": "",
        "activationCostCent": 0,
        "hourlyCostCent": 5000,
        "skills": ["alice"],
        "fixedStartTime": false,
        "maxWorkingTimeMin": 0,
        "origin": {
          "address": "2739 W Avenue L, Lancaster, CA 93536, USA",
          "lat": 34.66031,
          "lng": -118.17887,
          "geoAddress": "2739 West Avenue L, Lancaster, CA 93536, USA",
          "geoAccuracy": 8
        },
        "destination": {
          "address": "2739 W Avenue L, Lancaster, CA 93536, USA",
          "lat": 34.66031,
          "lng": -118.17887,
          "geoAddress": "2739 West Avenue L, Lancaster, CA 93536, USA",
          "geoAccuracy": 8
        },
        "timeWindow": {
          "startMin": 600,
          "stopMin": 1140
        },
        "speedPercent": 100,
        "maxCapacity": 8,
        "maxCapacity2": 80,
        "maxCapacity3": 0
      }
    }
  }],
  "orders": [{
    "name": "Nelson Torgersen",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "43534 Kirkland Ave, Lancaster, CA 93535, USA",
      "lat": 34.67427,
      "lng": -118.12454,
      "geoAddress": "43534 Kirkland Ave, Lancaster, CA 93535, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 1,
    "loadUnits2": 10,
    "loadUnits3": 0,
    "serviceTimeMin": 2,
    "importance": "normal",
    "customFields": {
      "orderID": "XY-1234",
      "customerID": "AAA-9999",
      "phoneNumber": "(987) 654-3210"
    },
    "eligibleDays": [1],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 540
    }]
  }, {
    "name": "Jamie Rozman",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "1027 W Newgrove St, Lancaster, CA 93534, USA",
      "lat": 34.69348,
      "lng": -118.14888,
      "geoAddress": "1027 W Newgrove St, Lancaster, CA 93534, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 2,
    "loadUnits2": 15,
    "loadUnits3": 0,
    "serviceTimeMin": 2,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [2],
    "timeWindows": [{
      "startMin": 375,
      "stopMin": 450
    }]
  }, {
    "name": "Jessie Blewett",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "1235 W Kettering St, Lancaster, CA 93534, USA",
      "lat": 34.69867,
      "lng": -118.15285,
      "geoAddress": "1235 W Kettering St, Lancaster, CA 93534, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 2,
    "loadUnits2": 18,
    "loadUnits3": 0,
    "serviceTimeMin": 2,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [1, 2],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 570
    }]
  }, {
    "name": "Marcie Ocon",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "45419 10th St W, Lancaster, CA 93534, USA",
      "lat": 34.70813,
      "lng": -118.14819,
      "geoAddress": "45419 10th St W, Lancaster, CA 93534, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 1,
    "loadUnits2": 7,
    "loadUnits3": 0,
    "serviceTimeMin": 2,
    "importance": "highest",
    "customFields": [],
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 390,
      "stopMin": 390
    }]
  }, {
    "name": "Earlene Saur",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "44924 5th St E, Lancaster, CA 93535, USA",
      "lat": 34.69958,
      "lng": -118.1215,
      "geoAddress": "44924 5th St E, Lancaster, CA 93535, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 2,
    "loadUnits2": 22,
    "loadUnits3": 0,
    "serviceTimeMin": 0,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 600
    }]
  }, {
    "name": "Clayton Hervey",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "425 E Ave H 11, Lancaster, CA 93535, USA",
      "lat": 34.70878,
      "lng": -118.12165,
      "geoAddress": "425 E Ave H 11, Lancaster, CA 93535, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 1,
    "loadUnits2": 4,
    "loadUnits3": 0,
    "serviceTimeMin": 2,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 540,
      "stopMin": 600
    }]
  }, {
    "name": "Earnestine Harnett",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "Earnestine Harnett\u0027s front door",
      "lat": 34.66239,
      "lng": -118.18351,
      "geoAddress": "42801-42999 30th St W, Lancaster, CA 93536, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 1,
    "loadUnits2": 7,
    "loadUnits3": 0,
    "serviceTimeMin": 0,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 660
    }]
  }, {
```

```
    "name": "Nelson Lufkin",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "43420 22nd St W, Lancaster, CA 93536, USA",
      "lat": 34.67345,
      "lng": -118.17017,
      "geoAddress": "43420 22nd St W, Lancaster, CA 93536, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 3,
    "loadUnits2": 38,
    "loadUnits3": 0,
    "serviceTimeMin": 0,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 480,
      "stopMin": 540
    }]
  }, {
    "name": "Jerri Ertl",
    "notes": "",
    "skillsInclude": [],
    "skillsExclude": [],
    "location": {
      "address": "44044 36th St W, Lancaster, CA 93536, USA",
      "lat": 34.68294,
      "lng": -118.19472,
      "geoAddress": "44044 36th St W, Lancaster, CA 93536, USA",
      "geoAccuracy": 8,
      "geoStatus": 0
    },
    "loadUnits": 3,
    "loadUnits2": 32,
    "loadUnits3": 0,
    "serviceTimeMin": 2,
    "importance": "normal",
    "customFields": [],
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 450,
      "stopMin": 510
    }]
  }],
  "routes": [{
    "driverID": 1,
    "day": 1,
    "driverIdx": 0,
    "stops": [{
      "type": "origin",
      "departureTimeSec": 30471,
      "nextStopDriveTimeSec": 129
    }, {
      "type": "order",
      "orderIdx": 6,
      "arrivalTimeSec": 30600,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 30600,
      "departureTimeSec": 30600,
      "nextStopDriveTimeSec": 1248
    }, {
      "type": "order",
      "orderIdx": 0,
      "arrivalTimeSec": 31848,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 31848,
      "departureTimeSec": 31968,
      "nextStopDriveTimeSec": 509
    }, {
      "type": "order",
      "orderIdx": 4,
      "arrivalTimeSec": 32477,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 32477,
      "departureTimeSec": 32477,
      "nextStopDriveTimeSec": 184
    }, {
      "type": "order",
      "orderIdx": 5,
      "arrivalTimeSec": 32661,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 32661,
      "departureTimeSec": 32781,
      "nextStopDriveTimeSec": 551
    }, {
      "type": "order",
      "orderIdx": 2,
      "arrivalTimeSec": 33332,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 33332,
      "departureTimeSec": 33452,
      "nextStopDriveTimeSec": 0
    }]
  }, {
    "driverID": 1,
    "day": 2,
    "driverIdx": 0,
    "stops": [{
      "type": "origin",
      "departureTimeSec": 21916,
      "nextStopDriveTimeSec": 1484
    }, {
      "type": "order",
      "orderIdx": 3,
      "arrivalTimeSec": 23400,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 23400,
      "departureTimeSec": 23520,
      "nextStopDriveTimeSec": 293
    }, {
      "type": "order",
      "orderIdx": 1,
      "arrivalTimeSec": 23813,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 23813,
      "departureTimeSec": 23933,
      "nextStopDriveTimeSec": 1147
    }, {
      "type": "order",
      "orderIdx": 8,
      "arrivalTimeSec": 25080,
      "idleTimeSec": 1920,
      "serviceStartTimeSec": 27000,
      "departureTimeSec": 27120,
      "nextStopDriveTimeSec": 669
    }, {
      "type": "order",
      "orderIdx": 7,
      "arrivalTimeSec": 27789,
      "idleTimeSec": 1011,
      "serviceStartTimeSec": 28800,
      "departureTimeSec": 28800,
      "nextStopDriveTimeSec": 0
    }]
  }]
}
```

4.3 Create a new Routeplan

Description:

Create a new Routeplan.

URL:

```
https://vrp.viamente.com/api/vrp/v2/routeplans
```

Formats:

json

HTTP Method(s):

POST

Input parameters:

Property	Type	Description	Def. Value
name*	String	User-assigned Routeplan name. Max 100 characters.	-
label	String	User-assigned Routeplan label text. Max 250 characters.	-
horizonDays	Integer	The planning horizon, in days, for this Routeplan. Cannot be greater than the maximum planning horizon allowed by the purchased Subscription Plan.	1
workingDayStartMin	Integer	Marks the beginning, in minutes since midnight, of the 24-hours working day for this Routeplan. See also Chapter 2.5.	0
driversSpeedPercent	Integer	This is a percentage value between 50 and 200 and is a modifier affecting all driving times. Set below 100 if your vehicles travel at a lower-than-average speed, above 100 if they travel at a higher-than-average speed. **Note**: driversSpeedPercent can also be used to model global traffic conditions. For local traffic conditions use trafficProfile.	100
trafficProfile	TrafficProfile	The traffic profile applied to this Routeplan. A traffic profile is defined by multiple regions, each one with its own traffic level. **Note:** driversSpeedPercent and trafficProfile are independently applied. See Table 7 for TrafficProfile object details.	-
loadUnitType	String	Define what a load unit is in this Routeplan. This does not affect the optimization process and is simply a string used when exporting route data to provide a name for the load units. Examples for this value are: pounds, gallons, cubic meters, passengers, crates, dollars, goods, etc.	"goods"
loadUnit2Type	String	Define what a load unit #2 is in this Routeplan. The same considerations enunciated for loadUnitType	"pounds"

		apply here.	
loadUnit3Type	String	Define what a load unit #3 is in this Routeplan. The same considerations enunciated for loadUnitType apply here.	"cubic feet"
subfleetID	String	Id of the subfleet to be included in this Routeplan. See Chapter 3 on how to get a list of available subfleets.	-
subfleetExcludeDrivers	Array of Integer	Optional. Filters Drivers in the selected subfleet by excluding those whose ID is in the array (see Table 1 for the Driver ID property). Defaults to all Drivers in the subfleet being included.	-
orders*	Array of Order	List of all Orders included in this Routeplan. See Table 4 at Paragraph 4.2 for Order object details.	-

Request body example:

```
{
  "name": "Lancaster deliveries",
  "label": "Your label text here",
  "horizonDays": 2,
  "driversSpeedPercent": 100,
  "trafficProfile": {
    "name": "Lancaster",
    "description": "",
    "timeWindows": [{
      "startMin": 420,
      "stopMin": 540
    }, {
      "startMin": 960,
      "stopMin": 1140
    }],
    "trafficRegions": [{
      "trafficLevelPercent": 50,
      "twTrafficLevelPercent": [85, 120],
      "region": [{
        "lat": 34.707610,
        "lng": -118.175640
      }, {
        "lat": 34.708033,
        "lng": -118.231430
      }, {
        "lat": 34.661028,
        "lng": -118.243618
      }, {
        "lat": 34.626004,
        "lng": -118.216323
      }, {
        "lat": 34.631512,
        "lng": -118.157100
      }, {
        "lat": 34.693356,
        "lng": -118.151950
      }, {
        "lat": 34.700130,
        "lng": -118.161220
      }]
    }]
  },
  "loadUnitType": "Goods",
  "loadUnit2Type": "Pounds",
  "subfleetID": "flt-11111111-2222-3333-4444-555555555555",
  "orders": [{
    "name": "Nelson Torgersen",
    "loadUnits": 1,
    "loadUnits2": 10,
    "serviceTimeMin": 2,
    "location": {
      "address": "43534 Kirkland Ave, Lancaster, CA 93535, USA"
    },
    "eligibleDays": [1],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 540
    }],
    "importance": "higher",
    "skillsInclude": ["plumbing", "powertools"],
    "skillsExclude": ["landscaping"],
    "customFields": {
      "orderID": "XY-1234",
      "customerID": "AAA-9999",
      "phoneNumber": "(987) 654-3210"
    }
  }, {
    "name": "Jamie Rozman",
    "loadUnits": 2,
    "loadUnits2": 15,
    "serviceTimeMin": 2,
    "location": {
      "address": "1027 W Newgrove St, Lancaster, CA 93534, USA"
    },
    "eligibleDays": [2],
    "timeWindows": [{
      "startMin": 375,
      "stopMin": 450
    }]
  }, {
    "name": "Jessie Blewett",
    "loadUnits": 2,
    "loadUnits2": 18,
    "serviceTimeMin": 2,
    "location": {
      "address": "1235 W Kettering St, Lancaster, CA 93534, USA"
    },
    "eligibleDays": [1, 2],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 570
    }]
  }, {
    "name": "Marcie Ocon",
    "importance": "highest",
    "loadUnits": 1,
    "loadUnits2": 7,
    "serviceTimeMin": 2,
    "location": {
      "address": "45419 10th St W, Lancaster, CA 93534, USA"
    },
    "timeWindows": [{
      "startMin": 390,
      "stopMin": 390
    }]
  }, {
    "name": "Earlene Saur",
    "loadUnits": 2,
    "loadUnits2": 22,
    "serviceTimeMin": 0,
    "location": {
      "address": "44924 5th St E, Lancaster, CA 93535, USA"
    },
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 600
    }]
  }, {
```

```
    "name": "Clayton Hervey",
    "loadUnits": 1,
    "loadUnits2": 4,
    "serviceTimeMin": 2,
    "location": {
      "address": "425 E Ave H 11, Lancaster, CA 93535, USA"
    },
    "timeWindows": [{
      "startMin": 540,
      "stopMin": 600
    }]
  }, {
    "name": "Earnestine Harnett",
    "loadUnits": 1,
    "loadUnits2": 7,
    "serviceTimeMin": 0,
    "location": {
      "address": "Earnestine Harnett's front door",
      "lat": 34.66239,
      "lng": -118.18351
    },
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 660
    }]
  }, {
    "name": "Nelson Lufkin",
    "loadUnits": 3,
    "loadUnits2": 38,
    "serviceTimeMin": 0,
    "location": {
      "address": "43420 22nd St W, Lancaster, CA 93536, USA"
    },
    "timeWindows": [{
      "startMin": 480,
      "stopMin": 540
    }]
  }, {
    "name": "Jerri Ertl",
    "loadUnits": 3,
    "loadUnits2": 32,
    "serviceTimeMin": 2,
    "location": {
      "address": "44044 36th St W, Lancaster, CA 93536, USA"
    },
    "timeWindows": [{
      "startMin": 450,
      "stopMin": 510
    }]
  }]
}
```

Return values:

Property	Type	Description
status	String	"OK" in case of success, "ERR" in case of failure.
id	String	System-assigned unique Routeplan ID.
warnings	Array of Warning	List warnings encountered while importing the Routeplan data. For a complete list of warnings see Paragraph 4.6

Response body example:

```
{
  "status": "OK",
  "id": "flt-11111111-2222-3333-4444-555555555555",
  "warnings": [{
    "code": "SUBFLEET_NOT_FOUND",
    "message": "Subfleet [flt-11111111-2222-3333-4444-555555555555] not found. Skipping subfleet, Routeplan will only include Orders.",
    "severity": "high"
  }]
}
```

4.4 Update an existing Routeplan

Description:

Rename and/or add Orders to an existing Routeplan.

URL:

`https://vrp.viamente.com/api/vrp/v2/routeplans/{planID}`

Formats:

json

HTTP Method(s):

PUT

Input parameters:

Property	Type	Description
name*	String	User-assigned Routeplan name. If specified replaces the existing name.
label	String	User-assigned Routeplan label text. Max 250 characters. If specified overwrites the existing label.
orders*	Array of Order	List of Orders to be added to the Routeplan. See Table 4 at Paragraph 4.2 for Order object details.

Request body example:

```
{
  "name": "Lancaster deliveries - Updated",
  "orders": [{
    "name": "Angela Edwards",
    "loadUnits": 2,
    "serviceTimeMin": 4,
    "location": {
      "address": "2610 W Norberry St, Lancaster, CA 93536, USA"
    },
    "eligibleDays": [1],
    "timeWindows": [{
      "startMin": 510,
      "stopMin": 540
    }],
    "importance": "higher",
    "skillsInclude": ["plumbing", "powertools"],
    "skillsExclude": ["landscaping"]
  }, {
    "name": "Laurie Miles",
    "loadUnits": 5,
    "serviceTimeMin": 7,
    "location": {
      "address": "44115 Gadsden Ave, Lancaster, CA 93534, USA"
    },
    "eligibleDays": [],
    "timeWindows": [{
      "startMin": 475,
      "stopMin": 540
    }]
  }]
}
```

Return values:

Property	Type	Description
status	String	"OK" in case of success, "ERR" in case of failure.
id	String	System-assigned unique Routeplan ID.
warnings	Array of Warning	List warnings encountered while updating the Routeplan. For a complete list of warnings see Paragraph 4.6

Response body example:

```
{
  "status": "OK",
  "id": "rpl-bcebe522-4e7e-4c0e-b365-d9d6f2b44d59",
  "warnings": [{
    "code": "IGNORED_EXCESS_ORDERS",
    "message": "The sum of the Orders in the request (2) and the Orders already in the Routeplan (49)
exceeds the Account limit of 50 Orders. Orders in excess of this limit have been discarded.",
    "severity": "high"
  }]
}
```

4.5 Delete a Routeplan

Description:

Delete a saved Routeplan (this action cannot be undone).

URL:

`https://vrp.viamente.com/api/vrp/v2/routeplans/{planID}`

Formats:

json

HTTP Method(s):

DELETE

Return values:

Property	Type	Description
status	String	"OK" in case of success, "ERR" in case of failure.
id	String	ID of the deleted Routeplan.

Response body example:

```
{
  "status": "OK",
  "id": "rpl-bcebe522-4e7e-4c0e-b365-d9d6f2b44d59"
}
```

4.6 Warning Messages

Warnings are added to the output when creating a new Routeplan or when updating an existing one if the input contains inconsistencies (e.g.: when specifying both **forceDriverID** and **skillsInclude** parameters for the same Order) and/or when the geocoding of one or more addresses requires special attention (e.g.: the given address string is an incomplete address, or it contains a postcode that does not match the one returned by the geocoding process, or the address is in an unsupported Country, etc.).

A Warning object is formatted as per the table below:

Table 11: Warning object properties

Property	Type	Description
code	String	One of the warning codes described below.
message	String	A human-readable message further describing the warning.
severity	String	low, medium or high
orderIdx	Integer	The 0-based index of the problematic Order in the input orders array. This field is present only if the warning refers to an Order.

Where "code" is one of the following:

- **SUBFLEET_NOT_FOUND**: The "subfleetID" input field references a subfleet that does not exist.
- **GEOCODING_WARNING**: The address is incomplete and/or the geocoding process might not have been accurate. After logging in to the Viamente Route Planner you should check the location of this address before optimizing the Routeplan.
- **GEOCODING_FAILURE**: The address could not be geocoded at all and/or is in an unsupported country. After logging in to the Viamente Route Planner you should manually fix it before optimizing the Routeplan.
- **INVALID_ORDER**: One or more items in the "orders" array are not Order objects.
- **INVALID_ORDERS_ARRAY**: The "orders" input field is incorrect (e.g.: not an array)
- **INVALID_LOCATION**: The "location" input field is missing or incorrect.
- **INVALID_TIME_WINDOW**: The "timeWindows" input field is missing or incorrect (e.g.: stopMin is before startMin).

- **INVALID_FORCE_DRIVER_ID**: The "forceDriverID" input field is incorrect (e.g.: it references a Driver ID that is not part of the given subfleet).
- **INVALID_IMPORTANCE**: The "importance" input field is invalid.
- **INVALID_SKILL_ARRAY**: One of the skills input fields is incorrect (e.g.: not an array)
- **INVALID_ELIGIBLE_DAYS_ARRAY**: The "eligibleDays" input field is incorrect (e.g.: not an array).
- **INVALID_ELIGIBLE_DAYS**: The "eligibleDays" input field contains invalid values that will be ignored.
- **INVALID_CUSTOM_FIELDS**: The "customFields" input field is invalid.
- **INVALID_LOADS_ARRAY**: The provided loadUnits values are invalid. E.g.: have mixed signs (some positive, some negative).
- **INVALID_PAIRING**: The "pairedDeliveryIdx" input field is invalid or used inappropriately. E.g.: references a non-existing Order OR references an Order that has already been referenced by some other Order OR is set by an Order that is itself referenced as a delivery by some other Order.
- **MISMATCHED_HORIZON_DAYS**: The specified "horizonDays" input field is less than the maximum day found in one or more Orders' eligibleDays.
- **IGNORED_SKILLS**: Both the "skillsInclude" and "skillsExclude" input parameters have been ignored because the "forceDriverID" was specified ("forceDriverID" overrides any skill setting).
- **IGNORED_CUSTOM_FIELDS**: More than 10 different custom field names have been specified in a single request and the over-quota fields are being ignored.
- **IGNORED_EXCESS_ORDERS**: The number of Orders specified in the input exceeds the Account limits. Orders in excess of that limit are ignored/discarded.
- **IGNORED_EXCESS_TIME_WINDOWS**: More than 2 time windows specified for an Order.
- **IGNORED_EXCESS_HORIZON_DAYS:** The value of the "horizonDays" input field exceeds the Account limits and will be lowered to match them.

5 Opening a saved Routeplan

A saved Routeplan can be directly opened in the Viamente Route Planner web application using the following URL syntax:

```
https://vrp.viamente.com/?plan={planID}
```

Where planID can be either a Routeplan ID (e.g.: "rpl-bcebe522-4e7e-4c0e-b365-d9d6f2b44d59") or a Routeplan name or partial name (e.g.: "Monday").

Some usage examples are:

```
https://vrp.viamente.com/?plan=rpl-bcebe522-4e7e-4c0e-b365-d9d6f2b44d59
```

```
https://vrp.viamente.com/?plan=monday
```

If a name is used instead of an ID and multiple Routeplans match the same name, the most recently updated one is loaded.

If no Routeplan matches the given planID, the Viamente Route Planner web application is still loaded but will immediately return a "Routeplan does not exist" error.

Note: If the user navigating to the above URL is not already logged in she will first be redirected to the login screen.

Appendix A Upgrading from API v1 to v2

This Appendix details what changes from version 1 of the API.

A.1 New features

API v2 introduces support for the following new features:

- Multiday planning
- An additional Load Capacity parameter
- Per-Driver and per-day Speed Factors
- A per-driver setting to flag the Driver as "required"
- Time-variant traffic regions

These changes, especially multiday planning, required some changes in the API format.

A.2 Updated API URL

The root API URL for v2 is at: http://vrp.viamente.com/api/vrp/v2

A.3 Driver Object changes

The Driver output-only object is the one more affected by the changes introduced with v2.

While in v1 all Driver constraints where 1st level fields of the Driver object, in v2 and with multiday planning all constraints are defined on a per-day basis under the dayToSettings map. This is necessary because the same Driver can have different constraints for each working day.

Moreover, V2 adds support for the following new Driver constraints:

- speedPercent: per-Driver and per-day speed factor
- maxCapacity3: a new additional (and optional) load constraint

See Table 1 at Chapter 3.2 for more details.

A.4 Order Object changes

The Order object format remains identical with respect to v1, adding support for the following new constraints:

- eligibleDays: an array defining, in a multiday planning scenario, on which days the Order can be served
- loadUnits3: a new additional (and optional) load constraint

See Table 4 at Chapter 4.2 for more details.

A.5 Route Object changes

The Route object format remains identical with respect to v1, adding support for the following new fields:

- day: the day assigned to the route
- loadUnits3: a new additional (and optional) load constraint

See Table 9 at Chapter 4.2 for more details.

A.6 Routeplan Object changes

The Routeplan input object adds support for the following new fields:

- horizonDays: the length, in days, of the planning horizon
- label: an additional (optional) user-defined text field used to further describe the Routeplan
- loadUnit3Type: the name of the unit assigned to the new additional load constraint
- workingDayStartMin: Marks the beginning, in minutes since midnight, of the 24-hours working day for the Routeplan. See also Chapter 2.5.

The format of the Driver, Order and Route objects contained in a Routeplan changes as described in the preceding Paragraphs.

See Chapter 4.2 for details.

A.7 Time Window changes

This change is only relevant for Routeplans where Drivers work across midnight and/or work for more than 24 hours.

Version 1 of the API did not support multi-day Routeplans and used "absolute" times to define Time Windows. It also allowed Time Windows spanning more than 24 hours.

Version 2 natively handles multi-day Routeplans and to do so introduces the concept of a 24-hour "working day". The 24-hour "working day" by default starts at 0:00 (matching the "solar" day), but can be set to start at a different time to accommodate Routeplans where Drivers work during the night (see also Chapter 2.5).

As a result, starting from version 2 of the API, Time Windows spanning more than 24 hours cannot be entered anymore and all times are indicated between 0:00 and 24:00 and are interpreted according to the Routeplan's workingDayStartMin (where times with a value less than workingDayStartMin are considered to be after-midnight).

In the examples below we assume that Drivers work across midnight and that the "working day" starts at 18:00.

Example of a 7-hour across-midnight Time Window spanning from 22:00 to 5:00 of the following day:

With API v1:

```
{
  "startMin": 1320,
  "stopMin": 1740
}
```

With API v2:

```
{
  "startMin": 1320,
  "stopMin": 300
}
```

NOTE: to improve backward compatibility, times with values higher than 1440 are still accepted by v2, but they are automatically modulo-ed by 1440 (1740 % 1440 = 300).

Example of a 3-hour after-midnight Time Window spanning from 1:00 to 4:00:

With API v1:

```
{
  "startMin": 1500,
  "stopMin": 1680
}
```

With API v2:

```
{
  "startMin": 60,
  "stopMin": 240
}
```

A.8 Vehicle Break changes

Vehicle breaks start and stop times follow the same logic outlined in Chapter A.7.