

## Table of Contents

1	Document Revision History .....	3
2	Introduction.....	4
2.1	Basic concepts .....	4
2.1.1	The API is entirely HTTP-based.....	4
2.1.2	The API is a RESTful resource.....	4
2.1.3	The format used for structured data exchange is JSON .....	4
2.1.4	SSL is supported (and highly recommended) .....	4
2.1.5	Parameters have certain expectations.....	4
2.1.6	Authentication.....	4
2.1.7	Concurrency.....	5
2.2	REST API Resources .....	5
2.2.1	Account Resource.....	5
2.2.2	Vehicle handling Resource .....	5
2.2.3	Optimization Resource .....	5
2.2.4	Matrix handling Resource.....	5
2.2.5	History Resource.....	5
2.3	Optimization and Matrix overview.....	5
2.4	HTTP Response Codes and Errors.....	6
2.4.1	HTTP Status Codes .....	6
2.4.2	Error Messages .....	6
2.4.3	Error Codes .....	6
3	Account Resource.....	8
3.1	Get Account Details .....	8
4	Vehicle handling Resource .....	9
4.1	List Vehicles .....	9
4.2	Create a Vehicle.....	10
4.3	Update a Vehicle.....	12
4.4	Delete a Vehicle.....	13
5	Matrix handling Resource.....	15
5.1	List Matrices .....	15
5.2	Create a Matrix.....	16

5.3	Delete a Matrix .....	17
6	Optimization Resource .....	18
6.1	Overview.....	18
6.2	HTTP Request Body format .....	18
6.2.1	Input object properties.....	18
6.2.2	Raw HTTP request body example.....	20
6.3	HTTP Response Body format .....	21
6.3.1	Output object properties.....	21
6.3.2	Warning Messages.....	23
6.3.3	Raw HTTP response body example .....	24
6.4	One-shot optimization.....	26
6.5	Matrix based optimization .....	27
7	History Resource.....	28
7.1	List all history entries.....	28
7.2	List history entries for a specific Matrix .....	30

## 1 Document Revision History

Date	Description
2012-04-10	Vehicle breaks and multiple waypoint Time Windows are now out of beta. Updated examples to include the new features. Added C# examples.
2012-02-15	Added important clarifications on how Time Windows work when applied to Vehicles and Waypoints (see Chapters 4.2 and 6.2.1)
2012-02-15	Added URLs pointing to useful external RESTful and JSON resources at Chapters 2.1.2 and 2.1.3
2012-01-02	"Waypoint include tags with OR behavior" is now out of beta
2012-01-02	Changed the URL syntax for the Update Vehicle API from <code>http://api.viamente.com/sub/rest/vehicle</code> to <code>http://api.viamente.com/sub/rest/vehicle/name/{vehicleName}</code> The new URL syntax allows renaming a vehicle and is more coherent with the rest of the API. The old URL will continue to be supported
2011-11-28	Better Vehicle name validation (Vehicle names cannot be path-like to avoid problems when accessing the resource via GET requests, see "Create a Vehicle" for details on forbidden characters)
2011-11-15	Fields linked to features currently in beta are now clearly marked as such (see Vehicle Resource and Optimization Resource Chapters)
2011-11-03	Added a per-vehicle setting to determine if breaks can be scheduled during waypoints' service time ("allowBreakService" input parameter)
2011-11-01	Added breaks start time information in the Optimization Response Body ("driveBreaks" and "serviceBreaks" output fields)
2011-10-24	Added warning messages about unreachable/unlinkable coordinates in the Optimization Response Body ("warnings" output field)
2011-10-12	Added the following new features (beta): <ul style="list-style-type: none"> <li>• support for Vehicle breaks (e.g.: lunch break constraints)</li> <li>• support for multiple Waypoint time windows (e.g.: 8:00-12:00, 13:00-18:00)</li> <li>• support Waypoint include tags with OR behavior (in addition to the existing AND behavior)</li> </ul>
2011-09-02	Added notes on concurrent access to the APIs and error 901
2011-08-18	Fixed minor typos
2011-06-01	Document created for API v2.0

## 2 Introduction

Viamente 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](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 both the authentication KEY and potentially confidential data (e.g.: client's geocoded locations) 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.

There are three special parameters in the API:

- **key**: Used for authentication, see section below for more details.
- **v**: used to request a specific API version. Currently ignored as the only version available is 2.0.
- **callback**: This parameter wraps your response in a callback method of your choice. For example, appending **callback=myFunction** to your request will result in a response body of: **myFunction( . . . )**. Callbacks may only contain alphanumeric characters and underscores; any invalid characters will be stripped.

#### 2.1.6 Authentication

Authentication is performed using an API key provided by Viamente. The API key must be included in the HTTP request using one of the following two methods:

- via the **key** query-string parameter;
- via the **X-Viamente-Key** HTTP header.

Where both are provided the parameter takes precedence.

The API key is a string formatted as AAAA-AAAA-AAAA-AAAA.

### 2.1.7 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" for the Optimization resource, that is: each API account can run one optimization at a time.

## 2.2 REST API Resources

The Viamente API includes the following RESTful resources to integrate with the Viamente optimization engine. Each resource is further detailed later in its own Chapter.

### 2.2.1 Account Resource

<http://api.viamente.com/sub/rest/account>

### 2.2.2 Vehicle handling Resource

<http://api.viamente.com/sub/rest/vehicle>

### 2.2.3 Optimization Resource

<http://api.viamente.com/sub/rest/optimization>

### 2.2.4 Matrix handling Resource

<http://api.viamente.com/sub/rest/matrix>

### 2.2.5 History Resource

<http://api.viamente.com/sub/rest/history>

## 2.3 Optimization and Matrix overview

The maximum allowed size of a single optimization request varies according to the size of the purchased subscription plan.

The size of an optimization request cannot exceed the Account fleet size and Route Plan size (in terms of the number of vehicles and waypoints included in the request). Note that multiple waypoints sharing the same coordinates count as their number and not as one and the same applies to vehicles.

There is no limit\* on the number of optimization requests that can be submitted, as long as their size and vehicle usage is allowed by the purchased subscription plan.

(\* fair usage policy applies. Too many or too frequent requests may be temporarily rejected with error code -900)

Optimizations can be "one-shot" or bound to a specific Matrix.

For a "one-shot" optimization the process is:

- 1) compute full driving duration Matrix against road network
- 2) optimize problem and discard Matrix

For an optimization bound to a Matrix the process changes as follows:

- 1) compute \*delta\* driving duration Matrix against road network
- 2) optimize problem
- 3) save updated Matrix

Bounding to a Matrix is especially useful when multiple optimizations are performed against the same or similar set of vehicles/waypoints. In such scenarios, at every new optimization request only a delta-matrix has to be computed thus greatly speeding up the whole process for quicker response times.

Note that the Matrix data is implicitly and automatically updated at every optimization request bound to that Matrix. Matrices can be created, listed and deleted but cannot be manipulated in any other way if not by the result of an optimization. A Matrix should therefore be considered as a simple “named token” to be used with the Optimization resource.

## 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:** Authentication credentials were missing or incorrect.
- **404 Not Found:** The URI requested is invalid or the resource requested does not exist.
- **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:** Viamente service is temporarily down or being upgraded.
- **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, it does so in JSON format (support for XML is being added). For example, an error might look like this:

```
{"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.
- **-200: Unknown Key:** Wrong authentication key.
- **-400: Key Constraint Violated:** One or more constraints associated with the authentication key, such as maximum number of Vehicles or Waypoints, have been violated.
- **-500: 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.
- **-600: Key Expired:** The authentication key is expired. Contact Viamente for a renewal.

- **-700: Unknown Matrix:** The specified Matrix does not exist.
- **-701: Matrix Already Exists:** The specified Matrix already exists.
- **-800: Matrix Locked:** The specified Matrix is currently in use by another optimization request. Please wait for such request to complete.
- **-850: Unknown Vehicle:** The specified Vehicle does not exist.
- **-851: Vehicle Already Exists:** The specified Vehicle already exists.
- **-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.
- **-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).

### 3 Account Resource

#### 3.1 Get Account Details

**Description:**

Return basic details about the Account.

**URL:**

`http://api.viamente.com/sub/rest/account`

**Formats:**

json

**HTTP Method(s):**

GET

**Return values:**

Property	Type	Description
<b>zone</b>	String	Indicates the account validity zone. Can be "EU" for European Union, "US" for United States, Canada and Mexico or "XE" for non-EU countries.
<b>creationDate</b>	Long	Account creation date in milliseconds (UNIX epoch).
<b>expiryDate</b>	Long	Account expiry date in milliseconds (UNIX epoch). Zero means no expiry.
<b>maxVehicles</b>	Integer	Maximum number of allowed Vehicles in each optimization request. Zero means unlimited.
<b>maxWaypoints</b>	Integer	Maximum number of allowed Waypoints in each optimization request. Zero means unlimited.

**Usage examples:**

```
curl http://api.viamente.com/sub/rest/account?key=AUTH_KEY
```

**Response body example:**

```
{
  "zone":"EU",
  "creationDate":1284569369000,
  "expiryDate":1287093600000,
  "maxVehicles":5,
  "maxWaypoints":100
}
```

## 4 Vehicle handling Resource

### 4.1 List Vehicles

**Description:**

List available Vehicles with their properties.

**URL:**

<http://api.viamente.com/sub/rest/vehicle>

**Formats:**

json

**HTTP Method(s):**

GET

**Return values:**

Property	Type	Description
<b>vehicles</b>	Array	Array of Vehicle objects.

Vehicle object properties:

Property	Type	Description
<b>name</b>	String	Vehicle name. Uniquely identify the vehicle.
<b>maxCapacity</b>	Integer	Maximum vehicle capacity in load units.
<b>timeWindow</b>	TimeWindow	If provided, defines the vehicle availability time range. If not specified, the vehicle will always be considered as available. See table at paragraph 6.2.1.2 for TimeWindow object properties.
<b>origin</b>	Location	The location of the start point for this vehicle. See table at paragraph 6.2.1.3 for Location object properties.
<b>destination</b>	Location	If provided, defines the location of the end-destination for this vehicle. If not specified, the route calculated for this vehicle terminates with the last visited waypoint. See table at paragraph 6.2.1.3 for Location object properties.
<b>tags</b>	Array of String	List of textual tags associated to this vehicle. Used together with waypoint's <b>tagsExclude</b> , <b>tagsIncludeAnd</b> and <b>tagsIncludeOr</b> fields, tags enable configuring complex vehicle-to-waypoint association constraints.
<b>breaks</b>	Array of Break	List of breaks defined for this vehicle. Defaults to an empty set. See table 4.2.1.1 for Break object properties.
<b>allowBreakService</b>	Boolean	If true breaks can be scheduled during waypoints' service time. If false breaks are only allowed during driving time. Defaults to false.

**Usage examples:**

curl http://api.viamente.com/sub/rest/vehicle?key=AUTH\_KEY

**Response body example:**

```
{
  "vehicles": [
    {
      "name": "Vehicle_A",
      "maxCapacity": 10,
      "timeWindow": {
        "startTimeSec": 36000,
        "stopTimeSec": 64800
      },
      "origin": {
        "latitude": 45.387404,
        "longitude": 9.260905
      },
      "tags": ["yellow"],
      "breaks": [{
        "startTimeSec": 43200,
        "stopTimeSec": 46800,
        "durationSec": 1800
      }],
      "allowBreakService": false
    },
    {
      "name": "Vehicle_B",
      "maxCapacity": 10,
      "timeWindow": {
        "startTimeSec": 36000,
        "stopTimeSec": 64800
      },
      "origin": {
        "latitude": 45.387404,
        "longitude": 9.260905
      },
      "destination": {
        "latitude": 45.387404,
        "longitude": 9.260905
      },
      "tags": ["black"]
    }
  ]
}
```

## 4.2 Create a Vehicle

**Description:**

Create a Vehicle to be used with the Optimization resource.

**URL:**

http://api.viamente.com/sub/rest/vehicle

**Formats:**

json

**HTTP Method(s):**

POST

**Input format:**

Fields marked with \* are mandatory.

Property	Type	Description	Def. Value
<b>name*</b>	String	Unique vehicle identifier. NOTE: The following substrings/characters are NOT allowed: '.', '/', '\', '&', '?', '#', '='	-
<b>maxCapacity</b>	Integer	Maximum vehicle capacity in load units.	1000
<b>timeWindow</b>	TimeWindow	If provided, defines the vehicle availability time range. If not specified, the vehicle will always be considered as available. The vehicle's time window will take into account the service time at the last visited waypoint and, if a destination is specified, the driving time to get there (that is, this API will never schedule a route so that the scheduled time of arrival at the last waypoint + the service time at that waypoint + the driving time to reach the destination exceeds the vehicle's time window). See table at paragraph 6.2.1.2 for TimeWindow object properties.	-
<b>origin*</b>	Location	The location of the start point for this vehicle. See table at paragraph 6.2.1.3 for Location object properties.	-
<b>destination</b>	Location	If provided, defines the location of the end-destination for this vehicle. If not specified, the route calculated for this vehicle terminates with the last visited waypoint. See table at paragraph 6.2.1.3 for Location object properties.	-
<b>tags</b>	Array of String	List of textual tags associated to this vehicle. Used together with waypoint's <b>tagsExclude</b> , <b>tagsIncludeAnd</b> and <b>tagsIncludeOr</b> fields, tags enable configuring complex vehicle-to-waypoint association constraints.	-
<b>breaks</b>	Array of Break	List of breaks defined for this vehicle. Defaults to an empty array. If multiple Breaks are specified, they must be non-overlapping. If a Time Window is specified for this vehicle then all defined breaks must be fully contained in the Time Window (i.e.: break's <code>startTimeSec</code> > <code>timewindow's startTimeSec</code> AND break's <code>stopTimeSec</code> + <code>durationSec</code> < <code>timewindow's stopTimeSec</code> ). See table below for Break object properties.	-
<b>allowBreakService</b>	Boolean	If true breaks can be scheduled during waypoints' service time. If false breaks are only allowed during driving time.	false

**4.2.1.1 Break object properties**

Property	Type	Description	Def. Value
<b>startTimeSec*</b>	Integer	Minimum allowed break start time in seconds since midnight (00:00). The allowed range can span a maximum of a 7-days period [0 – 604800].	-

<b>stopTimeSec*</b>	Integer	Maximum allowed break start time in seconds since midnight (00:00). The allowed range can span a maximum of a 7-days period [0 – 604800]	-
<b>durationSec*</b>	Integer	Break duration in seconds	-

**Return values:**

Return the same properties passed in the input.

**Usage examples:**

```
curl -d "@request.json" http://api.viamente.com/sub/rest/vehicle?key=AUTH_KEY
```

(where `request.json` is a local file containing the json data to be posted. See the request body example here below)

**Request body example:**

```
{
  "name": "Vehicle_A",
  "maxCapacity": 10,
  "timeWindow": {
    "startTimeSec": 36000,
    "stopTimeSec": 64800
  },
  "origin": {
    "latitude": 45.387404,
    "longitude": 9.260905
  },
  "tags": ["yellow"],
  "breaks": [{
    "startTimeSec": 43200,
    "stopTimeSec": 46800,
    "durationSec": 1800
  }],
  "allowBreakService": false
}
```

### 4.3 Update a Vehicle

**Description:**

Update an existing Vehicle to be used with the Optimization resource.

**URLs:**

```
http://api.viamente.com/sub/rest/vehicle/name/{vehicleName}
```

**Formats:**

json

**HTTP Method(s):**

PUT

**Input format:**

Same format as described in paragraph 4.2

**Return values:**

Return the same properties passed in the input.

**Usage examples:**

```
curl -X PUT -d "@request.json"  
http://api.viamente.com/sub/rest/vehicle/name/Vehicle_A?key=AUTH_KEY
```

(where `request.json` is a local file containing the json data to be posted. See the request body example here below)

**Request body example:**

```
{  
  "name": "Vehicle_A_renamed",  
  "maxCapacity": 15,  
  "timeWindow": {  
    "startTimeSec": 12000,  
    "stopTimeSec": 34200  
  },  
  "origin": {  
    "latitude": 45.387404,  
    "longitude": 9.260905  
  },  
  "destination": {  
    "latitude": 45.387404,  
    "longitude": 9.260905  
  },  
  "tags": [],  
  "breaks": [],  
  "allowBreakService": false  
}
```

## 4.4 Delete a Vehicle

**Description:**

Delete a Vehicle.

**URL:**

```
http://api.viamente.com/sub/rest/vehicle/name/{vehicleName}
```

**Formats:**

json

**HTTP Method(s):**

DELETE

**Usage examples:**

```
curl -X DELETE  
http://api.viamente.com/sub/rest/vehicle/name/Vehicle_A?key=AUTH_KEY
```

**Response body example:**

(empty response body)

## 5 Matrix handling Resource

### 5.1 List Matrices

**Description:**

List available Matrices with their name and ID.

**URL:**

`http://api.viamente.com/sub/rest/matrix`

**Formats:**

json

**HTTP Method(s):**

GET

**Return values:**

Property	Type	Description
<b>matrices</b>	Array	Array of Matrix objects.

Matrix object properties:

Property	Type	Description
<b>id</b>	Integer	Matrix ID.
<b>name</b>	String	Matrix name.
<b>creationDate</b>	Long	Creation date in milliseconds (UNIX epoch).

**Usage examples:**

`curl http://api.viamente.com/sub/rest/matrix?key=AUTH_KEY`

**Response body example:**

```
{
  "matrices": [
    {
      "id": 550,
      "name": "myFirstMtx",
      "creationDate": 1286446431000
    },
    {
      "id": 551,
      "name": "mySecondMtx",
      "creationDate": 1286446437000
    }
  ]
}
```

## 5.2 Create a Matrix

### Description:

Create an empty Matrix to be used with the Optimization resource.

### URL:

`http://api.viamente.com/sub/rest/matrix`

### Formats:

json

### HTTP Method(s):

POST

### Return values:

Property	Type	Description
<b>id</b>	Integer	Matrix ID.
<b>name</b>	String	Matrix name.
<b>creationDate</b>	Long	Creation date in milliseconds (UNIX epoch).

### Usage examples:

```
curl -X POST http://api.viamente.com/sub/rest/matrix?key=AUTH_KEY&name=myMtx
```

### Response body example:

```
{  
  "id":551,  
  "name":"myMtx",  
  "creationDate":1286455766735  
}
```

### 5.3 Delete a Matrix

**Description:**

Delete a Matrix.

**URLs:**

`http://api.viamente.com/sub/rest/matrix/id/{matrixID}`

`http://api.viamente.com/sub/rest/matrix/name/{matrixName}`

**Formats:**

json

**HTTP Method(s):**

DELETE

**Usage examples:**

```
curl -X DELETE http://api.viamente.com/sub/rest/matrix/id/551?key=AUTH_KEY
```

```
curl -X DELETE  
http://api.viamente.com/sub/rest/matrix/name/myMtx?key=AUTH_KEY
```

**Response body example:**

(empty response body)

## 6 Optimization Resource

### 6.1 Overview

Optimizes a set of waypoints so that they are best allocated to a set of vehicles taking into account the following complex constraints:

- vehicle limited capacity and working time-window;
- waypoint service time, delivery time-window and priority;
- custom exclusion/inclusion rules.

The optimization goals are defined by the following list of priorities:

- maximize number of serviced waypoints;
- minimize number of vehicles;
- minimize total working time (drive time + service time + idle time).

### 6.2 HTTP Request Body format

Requests to the Optimization Resource require a number of structured input parameters, all of which are listed below.

Fields marked with \* are mandatory.

#### 6.2.1 Input object properties

Property	Type	Description	Def. Value
<b>trafficFactor</b>	Float	Route duration multiplier. Used to take into account different traffic scenarios. Set to 1.0 for normal traffic conditions.	1.0
<b>balanced</b>	Boolean	Set to true to balance route duration, set to false to disable route duration balancing and minimize the overall duration of all routes.	true
<b>vehicleNames*</b>	Array of Vehicle names	Array of strings referencing vehicles previously created using the Vehicle resource (see Chapter 4 for details on how to create vehicles)	-
<b>waypoints*</b>	Array of Waypoint objects	Array of objects defining waypoint properties like latitude, longitude and delivery time window. See table below for Waypoint object properties.	-

##### 6.2.1.1 Waypoint object properties

Property	Type	Description	Def. Value
<b>name</b>	String	If provided, this name string can be used to identify this waypoint in the output.	-
<b>location*</b>	Location	Defines waypoint latitude/longitude coordinates.	-
<b>delivery</b>	Integer	Defines the amount of the delivery in terms of load units (i.e.: consumed vehicle capacity).	0
<b>serviceTimeSec</b>	Integer	The amount of time the vehicle is expected to be stationary at	0

		this waypoint in seconds (i.e.: time required for the servicing to take place).	
<b>timeWindow</b>	TimeWindow	Deprecated and replaced by "timeWindows". Supported for backward compatibility only.	-
<b>timeWindows</b>	Array of TimeWindow	If provided, defines an array of arrival time window constraints for this waypoint. Each time window defines a time interval when the waypoint 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 vehicle to arrive and start servicing the waypoint 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). Multiple time windows can be used, for example, to prevent the waypoint 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 waypoint from being serviced between 12:00 and 14:00). If not specified, this waypoint will always be available to be serviced. See table below for TimeWindow object properties.	(no constraint)
<b>priority</b>	Integer	Defines the priority of this waypoint where 0 is the lowest possible priority. NOTE: Priority is only taken into account when all Waypoints cannot be serviced by the given Vehicles: under such circumstances Waypoints with a higher priority are preferred over lower-priority ones.	0
<b>tagsInclude</b>	Array of String	Deprecated and replaced by "tagsIncludeAnd". Supported for backward compatibility only.	-
<b>tagsIncludeAnd</b>	Array of String	Define tag-based vehicle inclusion constraints: only vehicles defining <i>all tags in this list</i> (AND-criteria) are allowed to service this waypoint.	-
<b>tagsIncludeOr</b>	Array of String	Define tag-based vehicle inclusion constraints: all vehicles defining <i>one or more tags in this list</i> (OR-criteria) are allowed to service this waypoint.	-
<b>tagsExclude</b>	Array of String	Define tag-based vehicle exclusion constraints: all vehicles defining <i>one or more tags in this list</i> are <u>not</u> allowed to service this waypoint. <u>Exclusion constraints take precedence over inclusion ones.</u>	-

### 6.2.1.2 TimeWindow object properties

Property	Type	Description	Def. Value
<b>startTimeSec*</b>	Integer	Time window start time in seconds since midnight (00:00). The allowed range can span a maximum of a 7-days period [0 – 604800]	0
<b>stopTimeSec*</b>	Integer	Time window stop time in seconds since midnight (00:00).	86400

The allowed range can span a maximum of a 7-days period [0 – 604800]

### 6.2.1.3 Location object properties

Property	Type	Description	Def. Value
<b>latitude*</b>	Double	Latitude coordinate specified using the WGS 84 reference frame (e.g.: 45.397204).	-
<b>longitude*</b>	Double	Longitude coordinate specified using the WGS 84 reference frame (e.g.: 9.251765).	-

## 6.2.2 Raw HTTP request body example

```
{
  "balanced": true,
  "trafficFactor": 1.0,
  "vehicleNames": ["Vehicle_A", "Vehicle_B"],
  "waypoints": [
    {
      "name": "wp 0",
      "location": {
        "latitude": 45.426321,
        "longitude": 9.207807
      },
      "delivery": 3,
      "serviceTimeSec": 900,
      "timeWindows": [{
        "startTimeSec": 36000,
        "stopTimeSec": 43200
      }],
      "tagsIncludeAnd": ["yellow"]
    },
    {
      "name": "wp 1",
      "location": {
        "latitude": 45.428429,
        "longitude": 9.203758
      },
      "delivery": 4,
      "serviceTimeSec": 960,
      "timeWindows": [{
        "startTimeSec": 46800,
        "stopTimeSec": 50400
      }],
      "tagsExclude": ["black", "purple", "green"]
    },
    {
      "name": "wp 2",
      "location": {
        "latitude": 45.438795,
        "longitude": 9.19936
      },
      "delivery": 4,
      "serviceTimeSec": 1140,
      "timeWindows": [{
        "startTimeSec": 43200,
        "stopTimeSec": 50400
      }],
      "tagsIncludeOr": ["yellow", "black"]
    },
    {
      "name": "wp 3",
      "location": {
        "latitude": 45.443327,
        "longitude": 9.191732
      }
    }
  ]
}
```

```

    },
    "delivery": 5,
    "serviceTimeSec": 900,
    "timeWindows": [{
      "startTimeSec": 43200,
      "stopTimeSec": 50400
    }]
  },
  {
    "name": "wp 4",
    "location": {
      "latitude": 45.543327,
      "longitude": 9.291732
    },
    "delivery": 8,
    "serviceTimeSec": 900,
    "timeWindows": [{
      "startTimeSec": 50400,
      "stopTimeSec": 64800
    }]
  }
}
1
}

```

### 6.3 HTTP Response Body format

A successful response from the API is structured as specified below.

#### 6.3.1 Output object properties

Property	Type	Description
<b>matrixID</b>	Integer	ID of the Matrix. -1 for one-shot optimizations.
<b>elapsedSec</b>	Integer	Duration of the optimization process in seconds.
<b>routes</b>	Array of Route objects	List of computed routes, one for each used vehicle.
<b>unreachedWaypointNumbers</b>	Array of Integer	List of waypoints that, with the given constraints, could not be reached/serviced. Each element represents the zero-based index of a waypoint in the waypoint array provided in the input.
<b>unreachedWaypointNames</b>	Array of String	List of waypoints that, with the given constraints, could not be reached/serviced. Each element represents the name of a waypoint as provided in the input (or an empty string if not provided).
<b>unneededVehicleNumbers</b>	Array of Integer	List of unneeded vehicles. Each element represents the zero-based index of a vehicle in the vehicle array provided in the input.
<b>unneededVehicleNames</b>	Array of String	List of unneeded vehicles. Each element represents the name of a vehicle as provided in the input.
<b>warnings</b>	Array of CodedMessage	List of warnings defined by a numeric code and a descriptive message. Warnings are issued when, for example, the optimization request completes successfully but some geodesic distances are used instead of actual driving durations because one or more coordinates cannot be linked to the road network.

##### 6.3.1.1 Route object properties

Property	Type	Description
----------	------	-------------

<b>vehicleNumber</b>	Integer	The zero-based index of the vehicle in the vehicle array provided in the input.
<b>vehicleName</b>	String	The name of the vehicle as provided in the input.
<b>totDriveTimeSec</b>	Integer	Total driving time in seconds.
<b>totIdleTimeSec</b>	Integer	Total idle time in seconds (i.e.: sum of all waypoint idle times). Idle time happens when, as per schedule, the Vehicle arrives early at a waypoint and must wait, idle, for the waypoint's time window to open.
<b>totServiceTimeSec</b>	Integer	Total service time in seconds (i.e.: sum of all waypoint service times).
<b>totWorkTimeSec</b>	Integer	Total working time (i.e.: sum of driving time, idle time and service time).
<b>steps</b>	Array of RouteStep objects	Ordered list of route steps. Route steps include one origin, one or more waypoints, zero or one destination (route ends at the last visited waypoint if no destination is specified for the vehicle).

### 6.3.1.2 RouteStep object properties

Property	Type	Description
<b>stepNumber</b>	Integer	A zero-based counter indicating the step number in this vehicle's route.
<b>waypointNumber</b>	Integer	The zero-based index of the waypoint in the waypoint array provided in the input. For route steps that are not waypoints like origin and destination it is set to -1 and -2 respectively.
<b>waypointName</b>	String	The name of the waypoint as provided in the input (or an empty string if not provided). For route steps that are not waypoints like origin and destination it is set to "origin" and "destination" respectively.
<b>latitude</b>	Double	Latitude coordinate specified using the WGS 84 reference frame (e.g.: 45.397204).
<b>longitude</b>	Double	Longitude coordinate specified using the WGS 84 reference frame (e.g.: 9.251765).
<b>timeWindow</b>	TimeWindow	Delivery time window constraint for this route step (i.e.: the visited Waypoint's matching time window). Set to the Vehicle's working time window for origins and empty for the end destination.
<b>arrivalTimeSec</b>	String	Arrival time in seconds since midnight.
<b>idleTimeSec</b>	Integer	This field is > 0 if the arrival time is earlier than the time window start time (in which case the vehicle has to wait until the time window start time).
<b>serviceStartTimeSec</b>	String	Time when the goods are delivered, in seconds since midnight.
<b>departureTimeSec</b>	String	Departure time in seconds since midnight.
<b>nextStepDriveTimeSec</b>	Integer	Time in seconds required to reach the next route step. Always set to 0 for the last step.
<b>driveBreakTimeSec</b>	Integer	This field is > 0 if the driving time between the previous step and this step is interrupted by one or more Vehicle-defined Breaks (e.g.: lunch break) and amounts to the sum of the durationSec of all involved Breaks.

		Note that arrivalTimeSec and all related times already take Vehicle Breaks into account. This field is omitted if no Vehicle Breaks are defined in the input request.
<b>driveBreaks</b>	Array of RouteStepBreak	Lists all breaks occurring during the driving time between the previous step and this step. This field is omitted if no Vehicle Breaks are defined in the input request.
<b>serviceBreakTimeSec</b>	Integer	This field is > 0 if the service time at this step is interrupted by one or more Vehicle-defined Breaks (e.g.: lunch break) and amounts to the sum of the durationSec of all involved Breaks. Note that departureTimeSec already take Vehicle Breaks into account. Note that the optimization engine will try to avoid placing Breaks during a service time if at all possible. This field is omitted if no Vehicle Breaks are defined in the input request.
<b>serviceBreaks</b>	Array of RouteStepBreak	Lists all breaks interrupting the service time at this step. This field is omitted if no Vehicle Breaks are defined in the input request.
<b>cumulativeCapacity</b>	Integer	Vehicle capacity consumed up to this step.
<b>cumulativeDriveTimeSec</b>	Integer	Cumulative driving time in seconds.
<b>cumulativeIdleTimeSec</b>	Integer	Cumulative idle time in seconds.
<b>cumulativeWorkTimeSec</b>	Integer	Cumulative working time in seconds.
<b>cumulativeServiceTimeSec</b>	Integer	Cumulative service time in seconds.

### 6.3.1.3 RouteStepBreak object properties

Property	Type	Description
<b>breakTimeSec</b>	Integer	Break start time in seconds since midnight.
<b>durationSec</b>	Integer	Break duration in seconds.

### 6.3.1.4 CodedMessage object properties

Property	Type	Description
<b>code</b>	Integer	Integer code defining this type of message.
<b>message</b>	String	Message text.

### 6.3.2 Warning Messages

Warnings are issued when, for example, the optimization request completes successfully but some geodesic distances are used instead of actual driving durations because one or more coordinates cannot be linked to the road network.

The following warning codes may be returned:

- **-10000: Unlinkable coordinates:** Some coordinates (full list provided in message text) cannot be linked to the road network. Driving times to and from these locations are estimated using geodesic distances, which may lead to less optimal results and less accurate schedules.
- **-10100: Unreachable coordinates:** Some coordinates (full list provided in message text) are linked to restricted roads and cannot be reached. Driving times to these locations are estimated using geodesic distances, which may lead to less optimal results and less accurate schedules.

### 6.3.3 Raw HTTP response body example

```
{
  "matrixID": 551,
  "elapsedSec": 1,
  "routes": [{
    "vehicleNumber": 0,
    "vehicleName": "Vehicle_A",
    "totWorkTimeSec": 5282,
    "totIdleTimeSec": 773,
    "totServiceTimeSec": 1860,
    "totDriveTimeSec": 849,
    "steps": [{
      "stepNumber": 0,
      "waypointName": "origin",
      "waypointNumber": -1,
      "latitude": 45.387404,
      "longitude": 9.260905,
      "timeWindow": {
        "startTimeSec": 36000,
        "stopTimeSec": 64800
      },
      "arrivalTimeSec": 0,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 0,
      "departureTimeSec": 42478,
      "nextStepDriveTimeSec": 722,
      "cumulativeCapacity": 0,
      "cumulativeDriveTimeSec": 0,
      "cumulativeIdleTimeSec": 0,
      "cumulativeWorkTimeSec": 0,
      "cumulativeServiceTimeSec": 0
    }, {
      "stepNumber": 1,
      "waypointName": "wp 0",
      "waypointNumber": 0,
      "latitude": 45.426321,
      "longitude": 9.207807,
      "timeWindow": {
        "startTimeSec": 36000,
        "stopTimeSec": 43200
      },
      "arrivalTimeSec": 43200,
      "idleTimeSec": 0,
      "serviceStartTimeSec": 43200,
      "departureTimeSec": 44100,
      "nextStepDriveTimeSec": 127,
      "cumulativeCapacity": 3,
      "cumulativeDriveTimeSec": 722,
      "cumulativeIdleTimeSec": 0,
      "cumulativeWorkTimeSec": 1622,
      "cumulativeServiceTimeSec": 900
    }, {
      "stepNumber": 2,
      "waypointName": "wp 1",
      "waypointNumber": 1,
      "latitude": 45.428429,
      "longitude": 9.203758,
      "timeWindow": {
        "startTimeSec": 46800,
        "stopTimeSec": 50400
      },
      "arrivalTimeSec": 46027,
      "idleTimeSec": 773,
      "serviceStartTimeSec": 46800,
      "departureTimeSec": 47760,
      "nextStepDriveTimeSec": 0,
      "cumulativeCapacity": 7,
      "cumulativeDriveTimeSec": 849,
      "cumulativeIdleTimeSec": 773,
      "cumulativeWorkTimeSec": 5282,

```

```

    "cumulativeServiceTimeSec": 1860,
    "driveBreakTimeSec": 1800,
    "serviceBreakTimeSec": 0,
    "driveBreaks": [{
      "breakTimeSec": 44227,
      "durationSec": 1800
    }],
    "serviceBreaks": []
  ]}
}, {
  "vehicleNumber": 1,
  "vehicleName": "Vehicle_B",
  "totWorkTimeSec": 3715,
  "totIdleTimeSec": 0,
  "totServiceTimeSec": 2040,
  "totDriveTimeSec": 1675,
  "steps": [{
    "stepNumber": 0,
    "waypointName": "origin",
    "waypointNumber": -1,
    "latitude": 45.387404,
    "longitude": 9.260905,
    "timeWindow": {
      "startTimeSec": 36000,
      "stopTimeSec": 64800
    },
    "arrivalTimeSec": 0,
    "idleTimeSec": 0,
    "serviceStartTimeSec": 0,
    "departureTimeSec": 42432,
    "nextStepDriveTimeSec": 768,
    "cumulativeCapacity": 0,
    "cumulativeDriveTimeSec": 0,
    "cumulativeIdleTimeSec": 0,
    "cumulativeWorkTimeSec": 0,
    "cumulativeServiceTimeSec": 0
  }, {
    "stepNumber": 1,
    "waypointName": "wp 2",
    "waypointNumber": 2,
    "latitude": 45.438795,
    "longitude": 9.19936,
    "timeWindow": {
      "startTimeSec": 43200,
      "stopTimeSec": 50400
    },
    "arrivalTimeSec": 43200,
    "idleTimeSec": 0,
    "serviceStartTimeSec": 43200,
    "departureTimeSec": 44340,
    "nextStepDriveTimeSec": 184,
    "cumulativeCapacity": 4,
    "cumulativeDriveTimeSec": 768,
    "cumulativeIdleTimeSec": 0,
    "cumulativeWorkTimeSec": 1908,
    "cumulativeServiceTimeSec": 1140
  }, {
    "stepNumber": 2,
    "waypointName": "wp 3",
    "waypointNumber": 3,
    "latitude": 45.443327,
    "longitude": 9.191732,
    "timeWindow": {
      "startTimeSec": 43200,
      "stopTimeSec": 50400
    },
    "arrivalTimeSec": 44524,
    "idleTimeSec": 0,
    "serviceStartTimeSec": 44524,
    "departureTimeSec": 45424,
    "nextStepDriveTimeSec": 723,
    "cumulativeCapacity": 9,

```

```

    "cumulativeDriveTimeSec": 952,
    "cumulativeIdleTimeSec": 0,
    "cumulativeWorkTimeSec": 2992,
    "cumulativeServiceTimeSec": 2040
  }, {
    "stepNumber": 3,
    "waypointName": "destination",
    "waypointNumber": -2,
    "latitude": 45.387404,
    "longitude": 9.260905,
    "arrivalTimeSec": 46147,
    "idleTimeSec": 0,
    "serviceStartTimeSec": 0,
    "departureTimeSec": 0,
    "nextStepDriveTimeSec": 0,
    "cumulativeCapacity": 9,
    "cumulativeDriveTimeSec": 1675,
    "cumulativeIdleTimeSec": 0,
    "cumulativeWorkTimeSec": 3715,
    "cumulativeServiceTimeSec": 2040
  }
],
"unreachedWaypointNumbers": [4],
"unreachedWaypointNames": ["wp 4"],
"unneededVehicleNumbers": [],
"unneededVehicleNames": [],
"warnings": []
}

```

## 6.4 One-shot optimization

### Description:

Perform a one-shot optimization.

### URL:

<http://api.viamente.com/sub/rest/optimization>

### Formats:

json

### HTTP Method(s):

POST

### Usage examples:

```

curl -d "@request.json"
http://api.viamente.com/sub/rest/optimization?key=AUTH_KEY

```

```

curl -d "@request.json"
http://api.viamente.com/sub/rest/optimization?key=AUTH_KEY

```

(where `request.json` is a local file containing the json data to be posted. See chapter 6.2 for details)

### Response body example:

See chapter 6.3 for details.

## 6.5 Matrix based optimization

### Description:

Perform an optimization based on a Matrix.

### URLs:

```
http://api.viamente.com/sub/rest/optimization/id/{mtxID}
```

```
http://api.viamente.com/sub/rest/optimization/name/{mtxName}
```

### Formats:

json

### HTTP Method(s):

POST

### Usage examples:

```
curl -d "@request.json"  
http://api.viamente.com/sub/rest/optimization/id/551?key=AUTH_KEY
```

```
curl -d "@request.json"  
http://api.viamente.com/sub/rest/optimization/name/myMtx?key=AUTH_KEY
```

(where `request.json` is a local file containing the json data to be posted. See chapter 6.2 for details)

### Response body example:

See chapter 6.3 for details.

## 7 History Resource

### 7.1 List all history entries

**Description:**

List all history events in inverse chronological order starting with the most recent one.

The list is limited to 100 events.

**URL:**

<http://api.viamente.com/sub/rest/history>

**Formats:**

json

**HTTP Method(s):**

GET

**Return values:**

Property	Type	Description
<b>histories</b>	Array	Array of history objects.

History object properties:

Property	Type	Description
<b>matrixID</b>	Integer	ID of the matrix this history entry refers to. 0 if it does not refer to a matrix.
<b>operation</b>	String	Operation type. Can be one of the following: <ul style="list-style-type: none"> <li>ACCOUNT_CREATE: entry added via administrative means on account creation.</li> <li>ACCOUNT_UPDATE: entry added via administrative means on account data update.</li> <li>KEY_CREATE: entry added via administrative means on account creation.</li> <li>KEY_UPDATE: entry added via administrative means on key limits update.</li> <li>MATRIX_CREATE: entry added on Matrix creation.</li> <li>MATRIX_UPDATE: entry added by the optimization process. Precede "OPTIMIZE" entries based on a Matrix.</li> <li>MATRIX_DELETE: entry added on Matrix removal.</li> <li>OPTIMIZE: entry added when performing an optimization.</li> </ul>
<b>reqID</b>	String	Unique ID of the request. Currently available only for OPTIMIZE and MATRIX_UPDATE operations.
<b>result</b>	String	OK or FAIL.
<b>description</b>	String	Short description of the operation.
<b>reqDate</b>	Long	Operation request date in milliseconds (UNIX epoch).

**elapsedMillis** Integer Operation elapsed time in milliseconds.

**Usage examples:**

```
curl http://api.viamente.com/sub/rest/history?key=AUTH_KEY
```

**Response body example:**

```
{
  "histories": [
    {
      "matrixID": 0,
      "operation": "KEY_CREATE",
      "result": "OK",
      "description": "Creating new key: expiry [yyyyMMdd], maxVehicles [30], maxWaypoints [500]",
      "reqDate": 1286464409000,
      "elapsedMillis": 2
    },
    {
      "matrixID": 0,
      "operation": "ACCOUNT_CREATE",
      "result": "OK",
      "description": "Creating new account: email [name.surname@site.com], name [Name Surname], balance [1000]",
      "reqDate": 1286464408000,
      "elapsedMillis": 1
    }
  ]
}
```

## 7.2 List history entries for a specific Matrix

### Description:

List all history events involving a specific Matrix in inverse chronological order starting with the most recent one.

The list is limited to 100 events.

### URLs:

`http://api.viamente.com/sub/rest/history/id/{matrixID}`

`http://api.viamente.com/sub/rest/history/name/{matrixName}`

### Formats:

json

### HTTP Method(s):

GET

### Return values:

Same as those detailed in Paragraph 7.1.

### Usage examples:

```
curl http://api.viamente.com/sub/rest/history/id/551?key=AUTH_KEY
```

```
curl http://api.viamente.com/sub/rest/history/name/myMtx?key=AUTH_KEY
```

### Response body example:

```
{
  "histories": [
    {
      "matrixID": 551,
      "operation": "OPTIMIZE",
      "reqID": "20101007_121353_0_013",
      "result": "OK",
      "description": "Optimized with matrix: name[myMtx]",
      "reqDate": 1286446433000,
      "elapsedMillis": 203
    },
    {
      "matrixID": 551,
      "operation": "MATRIX_UPDATE",
      "reqID": "20101007_121353_0_013",
      "result": "OK",
      "description": "Updated matrix: added [1] locations, removed [0] locations",
      "reqDate": 1286446433000,
      "elapsedMillis": 86
    },
    {
      "matrixID": 551,
      "operation": "OPTIMIZE",
      "reqID": "20101007_121351_0_012",
      "result": "OK",
      "description": "Optimized with matrix: name[myMtx]",

```

```
"reqDate": 1286446431000,  
"elapsedMillis": 354  
},  
{  
  "matrixID": 551,  
  "operation": "MATRIX_UPDATE",  
  "reqID": "20101007_121351_0_012",  
  "result": "OK",  
  "description": "Updated matrix: added [5] locations, removed [0] locations",  
  "reqDate": 1286446431000,  
  "elapsedMillis": 92  
},  
{  
  "matrixID": 551,  
  "operation": "MATRIX_CREATE",  
  "result": "OK",  
  "description": "Created new matrix: name[myMtx], locations[0]",  
  "reqDate": 1286446431000,  
  "elapsedMillis": 2  
}  
]  
}
```

