Calculer un itinéraire

Request

Method	URL
POST	`https://backend.geovelo.fr/api/v2/computedroutes` *+queryParams* *+headerParams* *+bodyParams*

Query parameters

Key	Type	Values	Description
`instructions` **Optional**	boolean	true, false **Default**	Include the instructions in the response
`elevations` **Optional**	boolean	true, false **Default**	Include the elevations in the response
`geometry` **Optional**	boolean	true, false **Default**	Include the geometry in the response
`single_result` **Optional**	boolean	true, false **Default**	Expect only one route in the response instead of an array of route
`bike_stations` **Optional**	boolean	true **Default** , false	Include the bike stations near the start / end
`objects_as_ids` **Optional**	boolean	true **Default** , false	If true, returns only the ids of the objects. If false, returns the full objects.
`merge_instructions` **Optional**	boolean	false **Default** , true	If true, returns merge go straight forward instructions (This removes all instructions like "continue on road ..." due to intersections)
`show_pushing_bike_instructions` **Optional**	boolean	false **Default** , true	If true, it enable 2 news instructions to appear in instructions list : **GET_OFF_THE_BIKE** and **GET_ON_THE_BIKE**

Key	Type	Values	Description
			when the cyclist has to push its bike.

Header parameters

Key	Value
Content-Type Required	application/json
Source Optional	The source name given by Geovelo (when given)
Api-Key Required	The API key given by Geovelo

Body parameters

Key	Type	Values	Description
datetimeOfArrival Optional	string		Preferred datetime of arrival (format:Y-M-DTHH:MM:SS.zzz)
datetimeOfDeparture Optional	string		Prefered datetime of departure (format:Y-M-DTHH:MM:SS.zzz)
waypoints Required	Array<TWayPoint>		Ordered list of waypoints (see object description bellow)
bikeDetails Optional	TBikeDetails		See object description bellow
transportModes Optional	Array<'BIKE' | 'PEDESTRIAN'>	BIKE Default , PEDESTRIAN	Allowed transport modes

TWayPoint

Key	Type
latitude Required	number

Key	Type
longitude Required	number
title Required	string

TBikeDetails

Key	Type	Values	Description
profile Optional	string	BEGINNER, MEDIAN Default , EXPERT, VTC, CARGO	The bike profile to use to compute the itinerary. It sets the average speed accordingly
bikeType Optional	string	TRADITIONAL, BSS Default	The type of bike of the user
averageSpeed Required	number	13 km/h for BEGINNER Default , 16 km/h for MEDIAN, 20 km/h for EXPERT, 20 km/h for VTC, 15 km/h for CARGO	The average speed value must be between 5 and 45 km/h
bikeStations Optional	Array<TBikeStation>		Bike stations parameters
eBike Optional	boolean	true, false Default	Use an electric bike

TBikeStation

Key	Type	Values	Description
from Optional	number		The id of start bike station of departure
to Optional	number		The id of the bike station of arrival

Example :

JSON

```
POST  https://backend.geovelo.fr/api/v2/computedroutes?
instructions=false&elevations=false&geometry=true&singleResult=false

▼ {
    ▼ "waypoints" : [
        ▼ 0 : {
            "latitude" : 48.84102
            "longitude" : 2.320378
            "title" : "Gare Montparnasse"
        }
        ▼ 1 : {
            "latitude" : 48.858214
            "longitude" : 2.292516
            "title" : "Tour Eiffel"
        }
    ]
    ▼ "bikeDetails" : {
        "profile" : "MEDIAN"
        "bikeType" : "TRADITIONAL"
        "averageSpeed" : 15
        ▼ "bikeStations" : {
            "to" : 2894
            "from" : 3156
        }
```

Response

Status	Description
200 Ok	Returns an array of routes
400 Bad Request	Parameters sent are not valid

TRoute

Key	Type	Values	Description
id	string		The route id (can be used to retrieve

Key	Type	Values	Description
			more details)

`title`	`string`	`RECOMMENDED,`
		`SAFER,`
		`FASTER,`
		`BIS`
			An enum based title that the client has to localize

`distances`	`TDistance`		Contains all computed distances
`duration`	`number`		The duration in seconds
`waypoints`	`Array<TWayPoint>`		An ordered list of the points of the itinerary (start, step1, step2, ..., end)
`estimatedDatetimeOfDeparture`	`string`		Estimated datetime of departure
`estimatedDatetimeOfArrival`	`string`		Estimated datetime of arrival
`sections`	`Array<TSection>`		Contains all sections of the route

TDistances

Key	Type	Description
`normalRoads`	`number`	Distance of normal roads in meters
`recommendedRoads`	`number`	Distance of recommended roads in meters
`discouragedRoads`	`number`	Distance of discouraged roads in meters
`total`	`number`	Distance of all roads in meters

TWayPoint

Key	Type	Description
`latitude`	`number`	Latitude of the point
`longitude`	`number`	Longitude of the point
`title`	`string`	Title of the point (usually the address or some client-specific string : WORK, HOME, etc)

TSection

Key	Type	Values	Description
`duration`	`number`		The duration i seconds
`waypoints`	`Array<TWayPoint>`		Ordered list of waypoints
`estimatedDatetimeOfDeparture`	`string`		Estimated datetime of departure (format:Y-M-DTHH:MM:SS
`estimatedDatetimeOfArrival`	`string`		Estimated datetime of ar (format:Y-M-DTHH:MM:SS
`geometry`	`TGeometry`		The encoded geometry of th section
`waypointsIndices`	`number[]`		Waypoints ind in the geomet
`transportMode`	`string`	`BIKE` Default , `PEDESTRIAN`	Transport mo of the section
`details`	`TBikeDetails | TPedestrianDetails`		The details of section. This object can eit be a bikeDeta object or a pedestrianDet object depend on the transp mode of the section

TBikeDetails

Key	Type	Description
`profile`	`string`	Route profile for the section
`distances`	`TDistances`	Contains all computed distances for the section
`electric_adapted_itinerary`	`boolean`	True if the itinerary has been calculated of an electric bike, else False
`averageSpeed`	`number`	average speed for the section
`direction`	`string`	Main direction of the section
`bikeType`	`string`	Bike type of the section
`verticalGain`	`number`	Vertical gain (positive elevation) for the section
`instructions`	`Array<TInstruction>`	Contains all instructions
`elevations`	`Array<TElevation>`	Contains all elevations
`stepIndices`	`number[]`	Indices of itinerary steps in the geometry
`bikeStations`	`TBikeStations`	List of bike stations on the section

TPedestrianDetails

Key	Type	Description
`distances`	`TDistances`	Contains all computed distances for the section
`averageSpeed`	`number`	average speed for the section
`direction`	`string`	Main direction of the section
`verticalGain`	`number`	Vertical gain (positive elevation) for the section
`instructions`	`Array<TInstruction>`	Contains all instructions
`elevations`	`Array<TElevation>`	Contains all elevations
`stepIndices`	`number[]`	Indices of itinerary steps in the geometry

TInstruction

Key	Type		Description
direction	string	CROSSING, ELEVATOR, ENTER_AGAINST_ALLOWED_DIRECTION, ENTER_ROUND_ABOUT, new GET_OFF_THE_BIKE, new GET_ON_THE_BIKE, GO_STRAIGHT, HEAD_ON, LEAVE_AGAINST_ALLOWED_DIRECTION, LEAVE_ROUND_ABOUT, STAY_ON_ROUND_ABOUT, ROUND_ABOUT_EXIT_1, ROUND_ABOUT_EXIT_2, ROUND_ABOUT_EXIT_3, ROUND_ABOUT_EXIT_4, ROUND_ABOUT_EXIT_5, ROUND_ABOUT_EXIT_6, ROUND_ABOUT_EXIT_7, ROUND_ABOUT_EXIT_8, ROUND_ABOUT_EXIT_9, START_AT_END_OF_STREET, TAKE_SHARED_BIKE, DROP_SHARED_BIKE, TAKE_PUBLIC_TRANSPORT, LEAVE_PUBLIC_TRANSPORT, TURN_SLIGHT_RIGHT, TURN_RIGHT, TURN_SHARP_RIGHT, TURN_SHARP_LEFT, TURN_LEFT, TURN_SLIGHT_LEFT, REACHED_YOUR_DESTINATION, REACH_VIA_LOCATION, U_TURN, UNKNOWN	Instruction to follow (turn left, continue, etc.)
roadName	string		Name of the road to which the instruction leads
roadLength	number		Length of the road in meters
facility	string	CYCLEWAY, LANE, GREENWAY, OPPOSITE, SHAREBUSWAY, PEDESTRIAN, FOOTWAY, LIVINGSTREET, ZONE30, STEPS, FERRY,	Type of the road facililty

Key	Type		Description
		NONE, PRIMARY, SECONDARY, TERTIARY, RESIDENTIAL	
cyclability	number		Cyclability rating of the road (between 1 - 5)
geometryIndex	number		Index of the related point in the geometry field
orientation	string	E, N, NE, NW, S, SE, SW, W	Orientation to take to follow the instruction

In order to optimize network load and parsing, the instructions object follows the JSON HPack format :

```
▼ "instructions" : [
   ▼ [
       "field name #1"
       "field name #2"
       "..."
     ]
   ▼ [
       "instruction #1 - value of field #1"
       "instruction #1 - value of field #2"
       "..."
     ]
   ▼ [
       "instruction #2 - value of field #1"
       "instruction #2 - value of field #2"
       "..."
     ]
     "..."
  ]
```

Example :

```
▼ "instructions" : [
   ▼ 0 : [
       0 : "direction"
       1 : "roadName"
       2 : "roadLength"
```

```
            3 : "facility"
            4 : "cyclability"
            5 : "geometryIndex"
            6 : "orientation"
            7 : "cityNames"
        ]
      ▼ 1 : [
            0 : "HEAD_ON"
            1 : "Rue de Sambre et Meuse"
            2 : 48
            3 : "NONE"
            4 : 3
            5 : 0
            6 : "SE"
            7 : "Paris"
        ]
      ▼ 2 : [
            0 : "..."
        ]
      ▼ 3 : [
            0 : "REACHED_YOUR_DESTINATION"
            1 : ""
            2 : 0
            3 : "NONE"
            4 : 3
            5 : 12
            6 : "N"
            7 : ""
        ]
]
```

TGeometry

Key	Type	Description
`latitude`	`number`	Latitude of the point
`longitude`	`number`	Longitude of the point

```
// parse geometry polyline
import polyline from '@mapbox/polyline';
const geometry = polyline.toGeoJSON(_geometry, 6);
```

TElevation

Key	Type	Description
`distanceFromStart`	`number`	Total distance from section start
`elevation`	`number`	z-index on earth
`geometryIndex`	`number`	Index of the elevation in the section geometry

TBikeStations

Key	Type	Description
`to`	`TNearBikeStations`	The TNearBikeStations related to the itinerary start
`from`	`TNearBikeStations`	The TNearBikeStations related to the itinerary end

TNearBikeStations

Key	Type	Description
`near`	`number[]`	The list of BikeStations objects related to the itinerary start / end
`selected`	`number`	The id of the bike station used by this itinerary

Example :

JSON

```
POST  https://backend.geovelo.fr/api/v2/computedroutes?
instructions=true&elevations=false&geometry=true&single_result=false&bike_static
```

```
[
    0 : {
        "distances" : {
            "normalRoads" : 2044
            "recommendedRoads" : 838
            "total" : 2882
            "discouragedRoads" : 0
        }
        "estimatedDatetimeOfArrival" : "2016-09-12T18:23:29.127"
        "title" : "RECOMMENDED"
        "duration" : 858
        "sections" : [
            0 : {
```

```
"estimatedDatetimeOfArrival" :
"2016-09-12T18:10:17.127"
"geometry" : "aeif|AudtnCwAuF^[xJmIoKoCgQiH"
▼ "waypointsIndices" : [
    0 : 0
    1 : 6
]
▼ "details" : {
    "profile" : "WALKING"
    ▼ "distances" : {
        "opposite" : 0
        "normalRoads" : 93
        "recommendedRoads" : 0
        "discouragedRoads" : 0
        "greenway" : 0
        "total" : 93
        "lane" : 0
        "footway" : 10
        "livingstreet" : 0
        "pedestrian" : 57
        "residential" : 0
        "cycleway" : 0
        "steps" : 0
        "zone30" : 0
        "sharebusway" : 0
    }
    "direction" : "Boulevard de Magenta"
    "title" : "RECOMMENDED"
    "verticalGain" : 0
    ▼ "elevations" : [
        ▼ 0 : [
            0 : "distanceFromStart"
            1 : "elevation"
            2 : "geometryIndex"
        ]
        ▼ 1 : [
            0 : 0
            1 : 61.497202
            2 : 0
        ]
        ▼ 2 : [
            0 : "..."
        ]
        ▼ 3 : [
            0 : 93
            1 : 61.912394
            2 : 5
        ]
    ]
    ▼ "instructions" : [
        ▼ 0 : [
            0 : "direction"
        1 : "roadName"
        2 : "roadLength"
        3 : "facility"
        4 : "cyclability"
        5 : "geometryIndex"
        6 : "orientation"
        7 : "cityNames"
      ]
      ▼ 1 : [
        0 : "HEAD_ON"
        1 : "voie piétonne"
        2 : 10
        3 : "FOOTWAY"
        4 : 4
        5 : 0
        6 : "NE"
        7 : "Paris"
      ]
      ▼ 2 : [
        0 : "..."
      ]
      ▼ 3 : [
        0 : "REACHED_YOUR_DESTINATION"
        1 : ""
        2 : 0
        3 : "NONE"
        4 : 3
        5 : 5
        6 : "N"
        7 : ""
      ]
    ]
  }
  "duration" : 66
  "transportMode" : "PEDESTRIAN"
  "id" :
  "bG9jPTQ4Ljg3ODc0Mzc5ODUsMi4zNTMzOTE2NDczNCZsb2M9NDguODc5MDQ5MzEw
  "estimatedDatetimeOfDeparture" :
  "2016-09-12T18:09:11.127"
  ▼ "waypoints" : [
    ▼ 0 : {
      "latitude" : 48.87874379854197
      "longitude" : 2.3533916473388676
      "title" : NULL
    }
    ▼ 1 : {
      "latitude" : 48.87904931076023
      "longitude" : 2.354159095958248
      "title" : "10152 - GARE DU NORD 2"
    }
  ]
}
```

```
▼ 1 : {
    "estimatedDatetimeOfArrival" :
    "2016-09-12T18:20:25.127"
    "geometry" :
    "yyif|AoeunC{q@cYtDobAWyInFx@rUpDtJzA`Fv@`Er@te@nIbBXpBZbk@|IbBbEr
    yHEueBjw@q_AlSaVdi@on@w@hDaDbPaApE"
  ▼ "waypointsIndices" : [
      0 : 0
      1 : 73
    ]
  ▼ "details" : {
      "profile" : "MEDIAN"
    ▼ "distances" : {
        "opposite" : 0
        "normalRoads" : 1695
        "recommendedRoads" : 838
        "discouragedRoads" : 0
        "greenway" : 0
        "total" : 2533
        "lane" : 0
        "footway" : 0
        "livingstreet" : 0
        "pedestrian" : 188
        "residential" : 0
        "cycleway" : 838
        "steps" : 0
        "zone30" : 1007
        "sharebusway" : 0
      }
      "direction" : "Boulevard de Magenta"
      "bikeType" : "BSS"
    ▼ "bikeStations" : {
      ▼ "to" : {
        ▼ "near" : [
            0 : 794
          ]
          "selected" : 794
        }
      ▼ "from" : {
        ▼ "near" : [
            0 : 501
          ]
          "selected" : 501
        }
      }
      "verticalGain" : 19
      "averageSpeed" : 15
    ▼ "elevations" : [
      ▼ 0 : [
          0 : "distanceFromStart"
          1 : "elevation"
          2 : "geometryIndex"
        ]
```

```
          ▼ 1 : [
              0 : 0
              1 : 61.912394
              2 : 0
          ]
          ▼ 2 : [
              0 : "..."
          ]
          ▼ 3 : [
              0 : 2534
              1 : 63.317868
              2 : 72
          ]
      ]
      ▼ "instructions" : [
          ▼ 0 : [
              0 : "direction"
              1 : "roadName"
              2 : "roadLength"
              3 : "facility"
              4 : "cyclability"
              5 : "geometryIndex"
              6 : "orientation"
              7 : "cityNames"
          ]
          ▼ 1 : [
              0 : "TAKE_SHARED_BIKE"
              1 : ""
              2 : 0
              3 : "NONE"
              4 : 0
              5 : 0
              6 : "N"
              7 : ""
          ]
          ▼ 2 : [
              0 : "..."
          ]
          ▼ 3 : [
              0 : "DROP_SHARED_BIKE"
              1 : ""
              2 : 0
              3 : "NONE"
              4 : 0
              5 : 72
              6 : "N"
              7 : ""
          ]
      ]
  }
  "duration" : 608
  "transportMode" : "BIKE"
```

```
      "estimatedDatetimeOfDeparture" :
      "2016-09-12T18:10:17.127"
    ▼ "waypoints" : [
        ▼ 0 : {
            "latitude" : 48.87904931076023
            "longitude" : 2.354159095958248
            "title" : "10152 - GARE DU NORD 2"
          }
        ▼ 1 : {
            "latitude" : 48.87440566621717
            "longitude" : 2.373807237995119
            "title" : "10039 - SAMBRE ET MEUSE"
          }
      ]
    }
  ▼ 2 : {
      "estimatedDatetimeOfArrival" :
      "2016-09-12T18:23:29.127"
      "geometry" :
      "iw`f|Aih{oCdAqE`DcPv@iDrAyFuMzOk]`b@wJ{EqBy@mDmBiLyFaFyBiNkG"
    ▼ "waypointsIndices" : [
        0 : 0
        1 : 13
      ]
    ▼ "details" : {
        "profile" : "WALKING"
      ▼ "distances" : {
          "opposite" : 0
          "normalRoads" : 256
          "recommendedRoads" : 0
          "discouragedRoads" : 0
          "greenway" : 0
          "total" : 256
          "lane" : 0
          "footway" : 0
          "livingstreet" : 0
          "pedestrian" : 0
          "residential" : 0
          "cycleway" : 0
          "steps" : 0
          "zone30" : 0
          "sharebusway" : 0
        }
        "direction" : "Boulevard de la Villette"
        "title" : "RECOMMENDED"
        "verticalGain" : 4
      ▼ "elevations" : [
          ▼ 0 : [
              0 : "distanceFromStart"
              1 : "elevation"
              2 : "geometryIndex"
            ]
          ▼ 1 : [
        0 : 0
        1 : 63.315181
        2 : 0
      ]
      2 : [
        0 : "..."
      ]
      3 : [
        0 : 256
        1 : 70.925203
        2 : 12
      ]
    ]
    "instructions" : [
      0 : [
        0 : "direction"
        1 : "roadName"
        2 : "roadLength"
        3 : "facility"
        4 : "cyclability"
        5 : "geometryIndex"
        6 : "orientation"
        7 : "cityNames"
      ]
      1 : [
        0 : "HEAD_ON"
        1 : "Rue de Sambre et Meuse"
        2 : 48
        3 : "NONE"
        4 : 3
        5 : 0
        6 : "SE"
        7 : "Paris"
      ]
      2 : [
        0 : "..."
      ]
      3 : [
        0 : "REACHED_YOUR_DESTINATION"
        1 : ""
        2 : 0
        3 : "NONE"
        4 : 3
        5 : 12
        6 : "N"
        7 : ""
      ]
    ]
  }
  "duration" : 184
  "transportMode" : "PEDESTRIAN"
```

```
        "id" :
        "bG9jPTQ4Ljg3NDQwNTY2NjIsMi4zNzM4MDcyMzgmbG9jPTQ4Ljg3NTYzOTExOTMsM
        "estimatedDatetimeOfDeparture" :
        "2016-09-12T18:20:25.127"
        ▼ "waypoints" : [
            ▼ 0 : {
                "latitude" : 48.87440566621717
                "longitude" : 2.373807237995119
                "title" : "10039 - SAMBRE ET MEUSE"
            }
            ▼ 1 : {
                "latitude" : 48.87563911932326
                "longitude" : 2.3747634887695317
                "title" : NULL
            }
        ]
    }
```