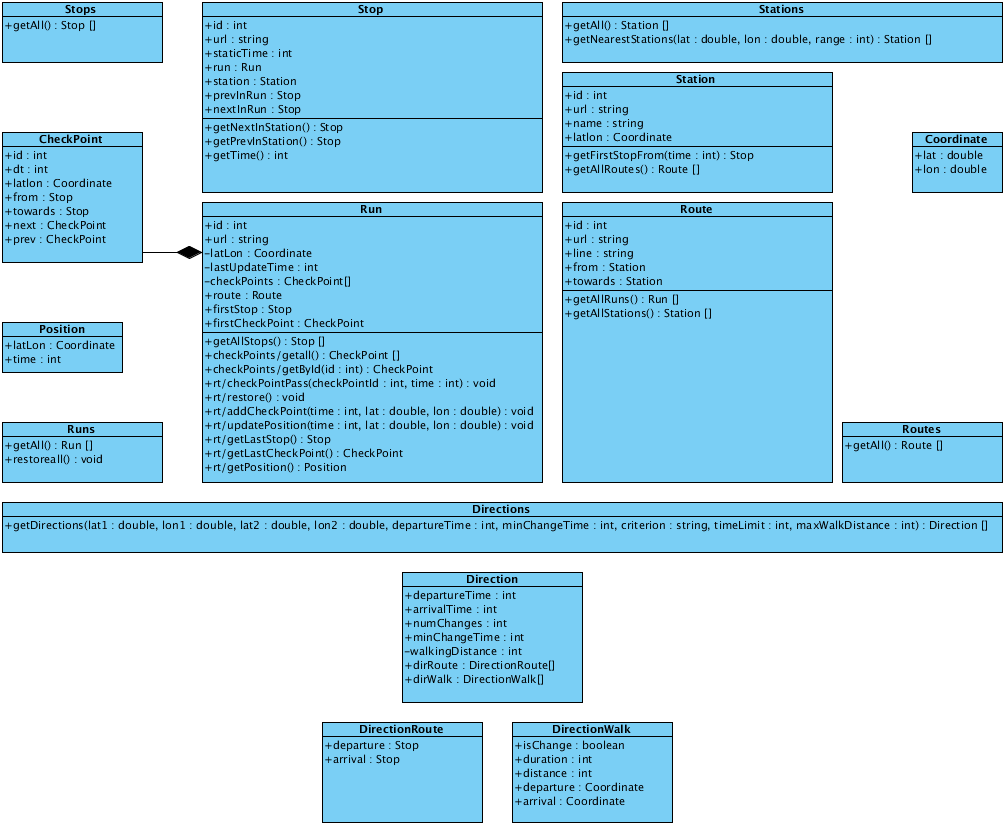
BusOnAir REST API Documentation

# 

# JSON Objects



## Details (VECCHIO! La nuova doc è su Mashape)

## (REST base path is: http://db.busonair.eu/plugin)

### Image1.png Stations

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /stations |

#### Operations

|  |  |
| --- | --- |
| public getAll () : List<Station> | |
| Rest Resource | /stations/getall |
| Comments | Result is sorted by id. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stationlist":["/stations/138","/stations/1","/stations/137","/stations/140","/stations/139"]} |

|  |  |
| --- | --- |
| public getNearestStations (lat : double, lon : double, [range : int]) : List<Station> | |
| Rest Resource | /stations/getneareststations?lat=924&lon=345&range=500 |
| Comments | Result is sorted by distance from point (lat,lon). |
| Parameters | |  | | --- | | lat |   Latitude   |  | | --- | | lon |   Longitude   |  | | --- | | range (optional) |   Range in meters (distance limit). If not specified all the stations will be returned. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stationlist":["/stations/2","/stations/53","/stations/51","/stations/99","/stations/1"]} |
| 400 | Lat and Lon cannot be blank. |

### Image1.png Stops

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /stops |

#### Operations

|  |  |
| --- | --- |
| public getAll () : List<Stop> | |
| Rest Resource | /stops/getall |
| Comments | Result is sorted by id. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stoplist":["/stops/16","/stops/15","/stops/12","/stops/11","/stops/14","/stops/13"]} |

### Image1.png Stop

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /stops/{id} |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"555","time":"708"} |
| 404 | No stop having the specified id value. |

#### Attributes

|  |  |
| --- | --- |
| public id : int | |
| Rest Resource | /stops/{id}/id |
| Id | |
| public time : int | |
| Rest Resource | /stops/{id}/time |
| Time in minutes from midnight. | |

#### Operations

|  |  |
| --- | --- |
| public getNextInRun () : Stop | |
| Rest Resource | /stops/{id}/getnextinrun |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"555","time":"708"} |
| 404 | No stop having the specified id value. |

|  |  |
| --- | --- |
| public getNextInStation () : Stop | |
| Rest Resource | /stops/{id}/getnextinstation |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"555","time":"708"} |
| 404 | No stop having the specified id value. |

|  |  |
| --- | --- |
| public getStation () : Station | |
| Rest Resource | /stops/{id}/getstation |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"555","time":"708"} |
| 404 | No stop having the specified id value. |

|  |  |
| --- | --- |
| public getRun () : Run | |
| Rest Resource | /stops/{id}/getrun |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"555","time":"708"} |
| 404 | No stop having the specified id value. |

### Image1.png Station

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /stations/{id} |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"latLon":{"lat":"42.34453","lon":"13.4011"},"stationId":"15","stationName":"terminalbus"} |
| 404 | No station having the specified id. |

#### Attributes

|  |  |
| --- | --- |
| public id : int | |
| Rest Resource | /stations/{id}/id |
| public name : string | |
| Rest Resource | /stations/{id}/name |
| public latlon : Coordinate | |
| Rest Resource | /stations/{id}/latlon |

#### Operations

|  |  |
| --- | --- |
| public getFirstStopFrom (time : int) : Stop | |
| Rest Resource | /stations/{id}/getfirststopfrom?time=520 |
| Parameters | |  | | --- | | time |   Time in minutes from midnight. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"1933","time":"955"} |
| 400 | Time cannot be blank. |
| 404 | No station having the specified id. |

|  |  |
| --- | --- |
| public getAllRoutes () : List<Route> | |
| Rest Resource | /stations/{id}/getallroutes |
| Comments | Result is sorted by id. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"routelist":["/routes/6","/routes/5","/routes/4","/routes/3"]} |
| 404 | No station having the specified id. |

### Image1.png Coordinate

#### Attributes

|  |
| --- |
| public lat : double |
| Latitude |
| public lon : double |
| Longitude |

### Image1.png Run

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /runs/{id} |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"runId":"2"} |
| 404 | No run having the specified id value. |

#### Attributes

|  |
| --- |
| public id : int |
| Id |

#### Operations

|  |  |
| --- | --- |
| public getRoute () : Route | |
| Rest Resource | /runs/{id}/getroute |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"routeId":"11","routeLine":"M12AD"} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public getFirstStop () : Stop | |
| Rest Resource | /runs/{id}/getfirststop |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stopId":"5425","time":"600"} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public getAllStops () : List<Stop> | |
| Rest Resource | /runs/{id}/getallstops |
| Comments | Result is sorted by the stop order in the run. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stoplist":["/stops/5431","/stops/5430","/stops/5429","/stops/5428","/stops/5427"]} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public getLastStop () : Stop | |
| Rest Resource | /runs/{id}/getlaststop |
| Comments | Return last visited stop. For runs that are not circulating (already arrived, or not yet started) last Stop of the Run will be returned. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"staticTime":"1110","stopId":"3539","time":"1110"} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public getLastCheckPoint() : CheckPoint | |
| Rest Resource | /runs/{id}/getlastcheckpoint |
| Comments | Return last visited checkpoint. For runs that are not circulating (already arrived, or not yet started) last CheckPoint of the Run will be returned. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"checkPointId":"24","from":"/stops/3539","latLon":{"lat":"42.34908","lon":"13.34783"},"next":"","time":"1110","towards":"/stops/3539"} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public checkpoints/getAll () : List<CheckPoint> | |
| Rest Resource | /runs/{id}/checkpoints/getall |
| Comments | Result is sorted by the checkpoint order in the run. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"checkpointlist":["/runs/3/checkpoints/1","/runs/3/checkpoints/2","/runs/3/checkpoints/3","/runs/3/checkpoints/4","/runs/3/checkpoints/0","/runs/3/checkpoints/10"]} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public checkpoints/getById(id : int) : CheckPoint | |
| Rest Resource | /runs/{idRun}/checkpoints/{id} |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"checkPointId":"2","from":"/stops/3528","latLon":{"lat":"42.34723","lon":"13.47286"},"next":"/runs/3/checkpoints/3","time":"1042","towards":"/stops/3528"} |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public update(checkPointId : int, time : int) | |
| Rest Resource | /runs/{id}/update?checkpointid=2&time=600 |
| Comments | Notifies that a circulating bus passed at the specified checkpoint. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | DONE |
| 400 | id, checkpointid, time cannot be blank |
| 404 | No run having the specified id. |

|  |  |
| --- | --- |
| public restore() | |
| Rest Resource | /runs/{id}/restore |
| Comments | Restore original time for each stop in the run. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | DONE |
| 404 | No run having the specified id. |

### Image1.png Route

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /routes/{id} |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"routeId":"11","routeLine":"M12AD"} |
| 404 | No route having the specified id value. |

#### Attributes

|  |  |
| --- | --- |
| public id : int | |
| Rest Resource | /routes/{id}/id |
| public line : string | |
| Rest Resource | /routes/{id}/line |

#### Operations

|  |  |
| --- | --- |
| public getAllRuns () : List<Run> | |
| Rest Resource | /routes/{id}/getallruns |
| Comments | Result is sorted by leaving time. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"runlist":["/runs/125","/runs/5243","/runs/12","/runs/8","/runs/41","/runs/97"]} |
| 404 | No route having the specified line value. |

|  |  |
| --- | --- |
| public getAllStations () : List<Station> | |
| Rest Resource | /routes/{id}/getallstations |
| Comments | Result is sorted by the stop order in the route. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"stationlist":["/stations/56","/stations/57","/stations/58","/stations/59","/stations/60"]} |
| 404 | No route having the specified line value. |

|  |  |
| --- | --- |
| public getFrom () : Station | |
| Rest Resource | /routes/{id}/getfrom |

#### Response

|  |  |  |  |
| --- | --- | --- | --- |
| HTTP State | Value | | |
| 200 | {"latLon":{"lat":"42.34453","lon":"13.4011"},"stationId":"15","stationName":"terminalbus"} | | |
| 404 | No route having the specified line value. | | |
| public getTowards() : Station | | |
| Rest Resource | | /routes/{id}/gettowards |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"latLon":{"lat":"42.33488","lon":"13.43763"},"stationId":"55","stationName":"s.elia-collevernesco"} |
| 404 | No route having the specified line value. |

### Image1.png Runs

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /runs |

#### Operations

|  |  |
| --- | --- |
| public getAll () : List<Run> | |
| Rest Resource | /runs/getall |
| Comments | Result is sorted by id. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"runlist":["/runs/0","/runs/1","/runs/2","/runs/3","/runs/4","/runs/5","/runs/6","/runs/7"]} |

### Image1.png Routes

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /routes |

#### Operations

|  |  |
| --- | --- |
| public getAll () : List<Route> | |
| Rest Resource | /routes/getall |
| Comments | Result is sorted by id. |

#### Response

|  |  |
| --- | --- |
| HTTP State | Value |
| 200 | {"routelist":["/routes/84","/routes/129","/routes/31","/routes/181","/routes/77"]} |

### Image1.png Directions

|  |  |
| --- | --- |
| Name | Value |
| Rest Resource | /directions |

#### Operations

|  |  |
| --- | --- |
| public getDirections (lat1 : double, lon1 : double, lat2 : double, lon2 : double, departureTime : int, minChangeTime : int, criterion : string, timeLimit : int, maxWalkDistance:int) : List<Direction> | |
| Rest Resource (example) | /getdirections? lat1=874&lon1=834&lat2=254&lon2=654&departuretime=520 &minchangetime=3&criterion=MINCHANGES |
| Comments | Result is the Pareto frontier of the paths for (arrival time, secondary criterion), sorted by arrival time of the founded paths (and reversely sorted by secondary criterion). |
| Parameters | |  | | --- | | lat1 |   Departure latitude   |  | | --- | | lon1 |   Departure longitude   |  | | --- | | lat2 |   Arrival latitude   |  | | --- | | lon2 |   Arrival longitude   |  | | --- | | departureTime |   Departure time in minutes from midnight.   |  | | --- | | minChangeTime |   Minimum time allowed for a bus change.   |  | | --- | | criterion |   Secondary criterion of optimization (first optimization criterion is always the arrival time). Allowed values:   * LATESTLEAVING: latest leaving path * SAFEST: safest changes path * MINCHANGES: minimum number of changes * MINWALK: minimum walk distance.  |  | | --- | | timeLimit |   Maximum duration allowed for the founded paths (in minutes). Default value is 600 (10 hours).   |  | | --- | | maxWalkDistance |   Maximum distance to walk to reach the departure station from the start point and the end point from the arrival station (in meters). Default value is 500. |

### Image1.png Direction

#### Attributes

|  |
| --- |
| public departureTime : int |
| Departure time in minutes from midnight. |
| public arrivalTime : int |
| Arrival time in minutes from midnight. |
| public numChanges : int |
| Number of expected bus changes in the path. |
| public minChangeTime : int |
| Minimum change time expected in the path. |
| public walkingDistance: int |
| Total distance (in meters) of the walks in the path. |
| public dirRoute : DirectionRoute [ ] |
| Bus directions. |
| public dirWalk : DirectionWalk [ ] |
| Walk directions. |

|  |  |
| --- | --- |
| Comments | A Direction must be read following the order:  DirectionWalk[0], DirectionRoute[0], DirectionWalk[1], DirectionRoute[1] ... |

### Image1.png DirectionRoute

#### Attributes

|  |
| --- |
| public departure : Stop |
| Departure of the bus direction. |
| public arrival : Stop |
| Arrival of the bus direction. |

|  |  |
| --- | --- |
| Comments | Stops between departure and arrival are not sent. Complete direction route (having all the stops) can be get by iteratively calling getnextinrun method on departure stop resource. |

### Image1.png DirectionWalk

#### Attributes

|  |
| --- |
| public isChange : boolean |
| True if the direction is a bus change. |
| public duration : int |
| Direction duration in minutes. |
| public distance : int |
| Direction distance in meters. |
| public departure : Coordinate |
| Departure of the walk direction. |
| public arrival : Coordinate |
| Arrival of the walk direction. |