

MultiRoute Webservice

Inhaltsverzeichnis

1	Webservice-Request	2
2	Webservice-Result	7
3	Webservice-StaticMap	12
4	Webservice-Status	14
5	Webservice-Optimierung	17
6	Postleitzahlen entlang der Tour	18
7	Zusätzliche Punkte hinzufügen	20
8	Webservice-Download	22
9	Weitere Aufrufe	24

gb consite GmbH
Mittenheimer Straße 58
85764 Oberschleißheim

Fon: +49-89-30905297-0
Fax: +49-89-30905297-9
E-Mail: info@gbconsite.de
Internet: www.gbconsite.de

Geschäftsführer
Malte Geschwinder
Knut Büscher

Sitz
Oberschleißheim

Amtsgericht München
HRB 168433

Ust-Id Nr.: DE254841920
Steuernr.: 143/139/70729

Bankverbindung
Kreissparkasse München Starnberg
BLZ 702 501 50
Konto 0022362933
IBAN: DE6970250150 0022362933
SWIFT-BIC: BYLADEM1KMS

1 Webservice-Request

1.1 Das Webservice-Request-Objekt

<https://webservice.tourenplaner.biz/startjob>

1.1.1 HTTP Methode: POST

1.1.2 Format / ContentType

Content-Type: application/json

1.2 Das JSON-POST-Object

1.2.1 "tour"

Parameter	Beschreibung	Werte
optimize	Optional Art der Optimierung	time(default) or distance
roundtrip	Optional Tourtype Rundstrecke oder nur Hinfahrt	true or false(default)
tollfree	Optional Tourparameter Vermeidung von Gebühren	true or false(default)
avoid_highway	Optional Tourparameter Vermeidung von Autobahnen	true or false(default)
uid	Optional CRM-uniqueID , mit welcher dieser Vorgang im CRM verwaltet wird.	String
start	Optional Startet automatisch die Routenoptimierung als Hintergrunddienst	true or false(default)
show_only	Optional Es wird keine Optimierung der Route vorgenommen (Tour nur anzeigen). Funktioniert nur bei start: true	true or false(default)
optimize_endpoint	Optional Der Endpunkt ist nicht fest, d.h. er wird auch in die Optimierung mit einbezogen. Funktioniert nur bei roundtrip: false	true or false(default)
time_window	Optional Zeitplanung mit Zeitfenster	true or false(default)
time_window_caclulation_time	Optional Maximale Zeit in Sekunden, die für die Berechnung der Zeitplanung mit Zeitfenster aufgewendet wird. Funktioniert nur bei time_window: true	1-600 (60) default

```
{
  "optimize": "time|distance",
  "roundtrip": true,
  "tollfree": true,
  "avoid_highway": true,
  "uid": "A123",
  "start": true,
  "show_only": true,
  "time_window": false
}
```



1.2.2 "waypoints"

Delivered must be an array with the individual waypoint object

1. The first point in the array is the start point
2. The last point is the end point if only the distance is to be calculated

Parameter	Beschreibung	Werte
uid	Optional CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist	String
address	Required Adresse	Object
coordinate	Optional Koordinaten des Wegpunktes, keine Geokodierung notwendig	Object
time	Optional Zeitangabe zu dem Wegpunkt*	Object
name	Optional Name der angezeigt werden soll	String

Es können beliebige key value (String) Paare angegeben werden, diese werden einfach durchgereicht.

Parameter address:

Parameter	Beschreibung	Werte
locality	Required Ort	Bsp.: München
postcode	Required Postleitzahl	Bsp.: 80331
street	Required Straße Hausnummer	Bsp.: Tal 1
country	Optional	Bsp.: DE, Deutschland oder Germany

Parameter coordinate:

Parameter	Beschreibung	Werte
lat	Required Latitude	Bsp.: 48.5
lon	Required Longitude	Bsp.: 11.5

Parameter time:

Parameter	Beschreibung	Werte
duration_of_stay	Required Aufenthaltszeit in Minuten	Bsp.: 60
arrival	Required Ankunftszeit	Bsp.: 2013-20-19T15:22:46+01:00
departure	Required Abfahrtszeit	Bsp.: 2013-20-19T15:22:46+01:00

Parameter time mit Zeitfenster:

Parameter	Beschreibung	Werte
duration_of_stay	Aufenthaltszeit in Minuten	Bsp.: 60
timeframe_start	Zeitfensterankunft	Bsp.: 2013-02-19T15:00:00+01:00
timeframe_end	Zeitfensterabfahrt (Der erste Punkt muss eine Abfahrtszeit haben)	Bsp.: 2013-02-19T16:00:00+01:00
timeframe_isfixed	Zeitfenster muss eingehalten werden	true oder false

Achtung

1. Zeitangaben in ISO 8601 Format
2. duration_of_stay wird nicht für den Ersten Punkt ausgewertet.
3. arrival, departure (bei Parameter **time**) es wird nur ein fixer Termin verwendet.
 - a. Es wird beim ersten Datensatz wo es eine arrival oder departure Angabe findet als Fixtermin festgelegt.

- b. Wenn arrival und departure angegeben sind, dann wird arrival als Fixtermin festgelegt.

Beispiel waypoints ohne Zeitfenster

```
[
  {
    "uid": "1",
    "coordinate" : {"lat": "48.163415", "lon": "11.577531"},
    "address": {"locality": "München", "postcode": "80803", "street":
"Pündterplatz 8"}
  },
  {
    "uid": "2",
    "coordinate" : {"lat": "49.454345", "lon": "11.073495"},
    "address": {"locality": "Nürnberg", "postcode": "90403", "street":
"Weintraubengasse 1"}
    "time": {"duration_of_stay" : 60, "arrival": "2013-20-
19T15:00:00+01:00"}
  },
  {
    "uid": "3",
    "coordinate" : {"lat": "48.136415", "lon": "11.577531"},
    "address": {"locality": "München", "postcode": "80331", "street":
"Tal 1"}
  }
]
```

Beispiel waypoints mit Zeitfenster

```
[
  {
    "uid": "1",
    "coordinate" : {"lat": "48.163415", "lon": "11.577531"},
    "address": {"locality": "München", "postcode": "80803", "street":
"Pündterplatz 8"},
    "time": {"timeframe_end": "2013-02-19T15:00:00+01:00"}
  },
  {
    "uid": "2",
    "coordinate" : {"lat": "49.454345", "lon": "11.073495"},
    "address": {"locality": "Nürnberg", "postcode": "90403", "street":
"Weintraubengasse 1"},
    "time": {"duration_of_stay" : 60, "timeframe_start": "2013-02-
19T15:00:00+01:00", "timeframe_end": "2013-02-19T18:00:00+01:00"}
  },
  {
    "uid": "3",
    "coordinate" : {"lat": "48.136415", "lon": "11.577531"},
    "address": {"locality": "München", "postcode": "80331", "street":
"Tal 1"}
  }
]
```

1.3 Webservice-Response-Objekt

1.3.1 Format / ContentType

Content-Type: application/json

1.3.2 Das JSON-Objekt

Parameter	Beschreibung	Werte
return_code	Status Code	Integer
error_description	Fehlerbeschreibung	String
tid	Transaktions-ID wenn erfolgreich	String
credit_expiration_date	Gültigkeit des Guthabens	DateTime

Beispiel

```
{
  "return_code": 0,
  "error_description": "",
  "tid": "bGVJKz6wFb-xilzKmWKpKw",
  "credit_expiration_date": "2014-02-26T14:00:02+01:00"
}
```

Parameter return code:

return_code	error_description
0	wie in UNIX: alles ok
2	Autorisierung: Abo abgelaufen
3	Inputdaten falsch oder unvollständig
5	Tour konnte nicht gespeichert werden
...	...

1.4 Beispielaufruf mit curl

```
curl -u test:test -XPOST -H "Content-Type: application/json" "https://web-  
service.tourenplaner.biz/startjob" -d '  
{  
  "tour": {  
    "roundtrip": true,  
    "tollfree": false,  
    "optimize": "time",  
    "avoid_highway": false,  
    "uid": "123",  
    "start": true  
  },  
  "waypoints": [  
    {  
      "uid": "34",  
      "coordinate" : {"lat": "48.163415", "lon": "11.577531"},  
      "address": {"locality": "München", "postcode": "80803", "street":  
"Pündterplatz 8"}  
    },  
    {  
      "uid": "26",  
      "coordinate" : {"lat": "49.454345", "lon": "11.073495"},  
      "address": {"locality": "Nürnberg", "postcode": "90403", "street":  
"Weintraubengasse 1"},  
      "time": {"duration_of_stay" : 60, "arrival": "2013-02-  
19T15:22:46+01:00"}  
    },  
    {  
      "uid": "56",  
      "coordinate" : {"lat": "48.136415", "lon": "11.577531"},  
      "address": {"locality": "München", "postcode": "80331", "street":  
"Tal 1"}  
    }  
  ]  
}'
```

2 Webservice-Result

2.1 Webservice-Result-Aufruf

<https://webservice.tourenplaner.biz/getresult?tid=<transaktions-ID>>

oder

<https://webservice.tourenplaner.biz/getresult.json?tid=<transaktions-ID>>

2.2 Das JSON-Object

2.2.1 "tour"

Parameter	Beschreibung	Werte
optimize	Art der Optimierung	time(default) or distance
roundtrip	Tourtype Rundstrecke oder nur Hinfahrt	true or false(default)
tollfree	Tourparameter Vermeidung von Gebühren	true or false(default)
avoid_highway	Tourparameter Vermeidung von Autobahnen	true or false(default)
uid	CRM-uniqueID , mit welcher dieser Vorgang im CRM verwaltet wird.	String
status	Zustand in dem sich die Tourenoptimierung befindet.	String
distance	Fahrstrecke in Meter	Integer
travel_time	Gesamtzeit in Sekunden (Fahrzeit + Aufenthaltsdauer)	Integer
driving_time	Fahrzeit in Sekunden	Integer
waiting_time	Wartezeit in Sekunden	Integer
time_window_error	Zeitfenster JSON Fehlerobjekt	null(default)

Beispiel

```
{
  "optimize": "time|distance",
  "roundtrip": true,
  "tollfree": true,
  "avoid_highway": true,
  "uid": "A123",
  "distance": 0,
  "travel_time": 0,
  "driving_time": 0,
  "waiting_time": 0
  "time_window_error": null
}
```

Parameter status:

status	Bedeutung
start	Daten wurden übergeben
geocoded	Alle Daten wurden Geokodiert
optimized	Tour wurde optimiert
scheduled	Zeitplanung wurde durchgeführt
scheduled_time_window	Zeitplanung mit Zeitfenstern wurde durchgeführt

Parameter time_window_error:

Zeitplanung mit Zeitfenster Fehler 1:

Ein Punkt konnte nicht vom Startpunkt aus erreicht werden (innerhalb des Zeitfensters).

```
"time_window_error": {
  "type": 1,
  "nr": 2,
  "name": "Punkt 3",
  "start_depo": "2:00 Uhr",
  "fahrzeit": "0h:10m",
  "ankunft_kunde": "2:10 Uhr",
  "timeframe_start": "1:30 Uhr",
  "timeframe_end": "1:45 Uhr"
}
```

Parameter	Bedeutung
time_window_error	
type	Fehlertyp 1: Ein Punkt in der Tour kann nicht vom Depot in der vorgeschriebenen Zeit erreicht werden.
nr	Aktuelle Position in den waypoints (Array beginnt mit 1)
name	CRM-uniqueID, unter welchem der Wegpunkt gespeichert ist
start_depo	Abfahrtszeit beim Depot
fahrzeit	Fahrzeit
ankunft_kunde	Ankunftszeit beim Kunden
timeframe_start	Zeitfenster für die Ankunft Anfang
timeframe_end	Zeitfenster für die Ankunft Ende

Zeitplanung mit Zeitfenster Fehler 2:

Zwischen 2 Punkten reicht die Zeit nicht, sie innerhalb einer Tour anzufahren.

```
"time_window_error": {
  "type": 2,
  "nr1": 2,
  "nr2": 3,
  "name1": "Name Punkt1",
  "name2": "Name Punkt2",
  "timeframe_start1": "3:30 Uhr",
  "timeframe_end1": "3:45 Uhr",
  "timeframe_start2": "3:30 Uhr",
  "timeframe_end2": "3:45 Uhr",
  "start_kunde1": "4:30 Uhr",
  "stop_kunde2": "6:12 Uhr",
  "start_kunde2": "4:30 Uhr",
  "stop_kunde1": "6:10 Uhr",
  "fahrzeit12": "1h:42m",
  "fahrzeit21": "1h:40m",
  "aufenthalt_kunde1": "1h:00m",
  "aufenthalt_kunde2": "1h:00m"
}
```



Parameter time_window_error	Bedeutung
type	Fehlertyp 2: Zwischen 2 Kunden gibt es keine Möglichkeit, sie innerhalb einer Tour anzufahren.
nr1	Aktuelle Position Kunde 1 in den waypoints (Array beginnt mit 1)
nr2	Aktuelle Position Kunde 2 in den waypoints (Array beginnt mit 1)
name1	CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist
name2	CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist
timeframe_start1	Zeitfenster für die Ankunft Anfang (Kunde 1)
timeframe_end1	Zeitfenster für die Ankunft Ende (Kunde 1)
timeframe_start2	Zeitfenster für die Ankunft Anfang (Kunde 2)
timeframe_end2	Zeitfenster für die Ankunft Ende (Kunde 2)
start_kunde1	Abfahrtszeit bei Kunde 1 (timeframe_start1 + aufenthalt_kunde1)
stop_kunde2	Ankunftszeit bei Kunde 2 (start_kunde1 + fahrzeit12)
start_kunde2	Abfahrtszeit beim Depot (timeframe_start2 + aufenthalt_kunde2)
stop_kunde1	Abfahrtszeit beim Depot (start_kunde2 + fahrzeit21)
fahrzeit12	Fahrzeit von Kunde 1 zu Kunde 2
fahrzeit21	Fahrzeit von Kunde 2 zu Kunde 1
aufenthalt_kunde1	Aufenthaltsdauer bei Kunde 1
aufenthalt_kunde2	Aufenthaltsdauer bei Kunde 2

2.2.2 "waypoints"

Geliefert werden muss ein Array mit den einzelnen Wegpunkt Objekt

1. Der erste Punkt im Array ist der Startpunkt
2. Der letzte Punkt ist der Endpunkt, wenn nur Hinweg berechnet wurde

Parameter	Beschreibung	Werte
uid	CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist	String
address	Adresse	Object
coordinate	Koordinaten des Wegpunktes	Object
time	Zeitangabe zu dem Wegpunkt*	Object
distance	Entfernung zum vorherigen Punkt	Integer
driving_time	Fahrzeit zum vorherigen Punkt	Integer

Achtung

Wenn der Parameter roundtrip: true (Rundreise) ist, dann beziehen sich die Parameter distance und driving_time auf die Zeit und Strecke, die vom letzten Punkt (letzter Punkt vor dem Ziel) zum Ersten (Start/Ziel) benötigt wird.

Parameter address (wird nicht verändert):

Parameter	Beschreibung	Werte
locality	Ort	Bsp.: München
postcode	Postleitzahl	Bsp.: 80331
street	Straße Hausnummer	Bsp.: Kaufinger Straße 5
country	Land	Bsp.: DE (wird nur ausgegeben wenn ursprünglich mit angegeben)

**Parameter coordinate:**

Parameter	Beschreibung	Werte
lat	Latitude	Bsp.: 48.5
lon	Longitude	Bsp.: 11.5

Parameter time:

Parameter	Beschreibung	Werte
duration_of_stay	Aufenthaltszeit in Minuten	Bsp.: 60
arrival	Ankunftszeit	Bsp.: 2013-20-19T15:22:46+01:00
departure	Abfahrtszeit	Bsp.: 2013-20-19T15:22:46+01:00
timeframe_start	Zeitfensterankunft	Bsp.: 2013-02-19T15:22:46+01:00
timeframe_end	Zeitfensterabfahrt	Bsp.: 2013-02-19T15:22:46+01:00
waiting_time	Wartezeit in Sekunden	Integer

Achtung

1. Zeitangaben in ISO 8601 Format
2. timeframe_start, timeframe_end, waiting_time werden nur ausgegeben, wenn auch Zeitfenster angegeben worden sind.

Beispiel

```
curl -u test:test -H "Content-Type: application/json" https://web-service.tourenplaner.biz/getresult?tid=
```

```
{
  "tour": {
    "roundtrip": true,
    "tollfree": false,
    "optimize": "time",
    "avoid_highway": false,
    "uid": "123",
    "status": "scheduled"
  },
  "waypoints": [
    {
      "uid": "34",
      "address": {
        "locality": "München",
        "postcode": "80803",
        "street": "Pündterplatz 8"
      },
      "coordinate": {
        "lat": 48.163415,
        "lon": 11.577531
      },
      "times": {
        "arrival": "2013-02-19T19:33:13+01:00",
        "departure": "2013-02-19T15:11:24+01:00"
      }
    },
    {
      "uid": "56",
      "address": {
        "locality": "München",
        "postcode": "80331",

```

```
    "street": "Tal 1"
  },
  "coordinate": {
    "lat": 48.136415,
    "lon": 11.577531
  },
  "times": {
    "arrival": "2013-02-19T15:22:46+01:00",
    "departure": "2013-02-19T15:22:46+01:00",
    "duration_of_stay": null,
    "timeframe_start": "2013-02-19T15:21:46+01:00",
    "timeframe_end": "2013-02-19T15:23:46+01:00",
    "waiting_time": 0
  }
},
{
  "uid": "26",
  "address": {
    "locality": "Nürnberg",
    "postcode": "90403",
    "street": "Weintraubengasse 1"
  },
  "coordinate": {
    "lat": 49.454345,
    "lon": 11.073495
  },
  "times": {
    "arrival": "2013-02-19T17:01:35+01:00",
    "departure": "2013-02-19T18:01:35+01:00",
    "duration_of_stay": 60,
    "timeframe_start": null,
    "timeframe_end": null,
    "waiting_time": 0
  }
}
]
}
```

3 Webservice-StaticMap

Statische Karte kann mit Route `/get_static_map?tid=xxx` angefordert werden.

3.1 Ausgabe-Steuerung

durch JSON-Objekt im POST-Request. Beispiel-JSON:

```
{
  "static_map": {
    "file_name": "mein_image.jpg",
    "disposition": "attachment",
    "mapsize_width": 800,
    "mapsize_height": 800
  }
}
```

3.1.1 zulässige Werte / Beschränkung

mapsize_width 80 - 900 (Pixel)

mapsize_height 80 - 834 (Pixel)

disposition "attachement" oder "inline" ("Download/Speichern"-Header oder normaler Header zum Gebrauch auf Website)

3.1.2 Default-Werte

file_name "staticmap.jpg"

disposition "attachement"

mapsize_width 400

mapsize_height 400

3.2 Fehlerfall

Im Fehlerfall wird ein JSON-Objekt der Form

```
{code: xx, description: 'yy'}
```

ausgeliefert.

code	description
11	Mehr als 100 Wegepunkte. Statische Karte kann nicht erstellt werden.
12	Keine Wegepunkte gefunden. Statische Karte kann nicht erstellt werden.
13	Schwerwiegender Fehler bei Kartenrequest.
14	Parameter mapsize_width muss zwischen 80 und 900 liegen (übergeben ...)

3.3 Erfolgsfall

Konnte die Karte erstellt werden, wird sie mit HTTP-Statuscode 200 und Header "image/jpeg" sowie weiteren Headern - abhängig von Parameter "disposition" - als JPEG-Image ausgeliefert. In diesem Fall ist keine JSON-Ausgabe möglich.

Zwischen 1 und 25 Wegepunkten werden Standardmarker und Route angezeigt, über 25 Punkten werden lediglich durchnummerierte Marker angezeigt.

3.4 vor Verwendung

Es ist vor Verwendung zu prüfen, ob der MIME-Type "image/jpg" ist. Wenn nicht, enthält die Rückgabe kein Image, sondern das Fehler-JSON.

3.5 Beispielaufruf mit Curl

tid=xxx ersetzen durch gültige tid. Die Ausgabe von curl wird in eine Datei umgeleitet, ansonsten sieht man im Erfolgsfall nur "binary"-Code auf dem Screen.

```
curl -u test:test -XPOST -H "Content-Type: application/json" "https://web-  
service.tourenplaner.biz/get_static_map?tid=xxx" -d '  
{  
  "static_map": {  
    "file_name": "mein_image.jpg",  
    "disposition": "attachment",  
    "mapsize_width": 800,  
    "mapsize_height": 800  
  }  
}
```



4 Webservice-Status

4.1 Webservice-Status-Aufruf

<https://webservice.tourenplaner.biz/getstatus?tid=<transaktions-ID>>

(Achtung Header "Content-Type: application/json" muss mit übergeben werden)
oder

<https://webservice.tourenplaner.biz/getstatus.json?tid=<transaktions-ID>>

4.2 Das JSON-Object

Parameter	Beschreibung	Werte
status	Zustand in dem sich die Tourenoptimierung befindet.	String
error_text	Fehlermeldung	null(default)
time_window_error	Zeitfenster JSON Fehlerobjekt	null(default)

4.2.1 Parameter status

status	Bedeutung
start	Daten wurden übergeben
geocoded	Alle Daten wurden Geokodiert
matrix	Fahrzeitmatrix wurde ermittelt
optimized	Tour wurde optimiert
scheduled	Zeitplanung wurde durchgeführt
scheduled_time_window	Zeitplanung mit Zeitfenstern wurde durchgeführt
error	Fehler ist während der Verarbeitung aufgetreten

4.2.2 Parameter error_text

error_text	Bedeutung	Lösung
Error geocoding: uid=xxx	Adresse mit der uid xxx konnte nicht geokodiert werden	Adresse herausnehmen.
Error matrix: uid=xxx	Adresse mit der uid xxx konnte nicht angefahren werden.	Adresse herausnehmen.
Optimisation not possible	Optimierung nicht möglich	Zeitfenster anpassen
Optimisation timeout	Optimierung mit Zeitfenster timeout	Tour Optimierung nochmals starten mit Parameter time_window_calculation_time
timeout	Fahrzeitmatrixermittlung hat zu lange gebraucht	Tour Optimierung nochmals starten
Unable to access network	Es gab Probleme mit der Applikation beim Aufruf der Geokodierung bzw. Fahrzeitmatrixermittlung	gb consite benachrichtigen
Internal Server Error	Es gab Probleme mit der Applikation beim Aufruf der Geokodierung bzw. Fahrzeitmatrixermittlung	gb consite benachrichtigen

4.2.3 Parameter time_window_error

JSON Objekt mit Fehlerstruktur zur Fehlermeldung (Zeitplanung mit Zeitfenstern).

Zeitplanung mit Zeitfenster Fehler 1:

Ein Punkt konnte nicht vom Startpunkt aus erreicht werden (innerhalb des Zeitfensters).

```
"time_window_error": {
  "type": 1,
  "nr": 2,
  "name": "Punkt 3",
  "start_depo": "2:00 Uhr",
  "fahrzeit": "0h:10m",
  "ankunft_kunde": "2:10 Uhr",
  "timeframe_start": "1:30 Uhr",
  "timeframe_end": "1:45 Uhr"
}
```

Parameter	Bedeutung
time_window_error	
type	Fehlertyp 1: Ein Punkt in der Tour kann nicht vom Depot in der vorgeschriebenen Zeit erreicht werden.
nr	Aktuelle Position in den waypoints (Array beginnt mit 1)
name	CRM-uniqueID, unter welchem der Wegpunkt gespeichert ist
start_depo	Abfahrtszeit beim Depot
fahrzeit	Fahrzeit
ankunft_kunde	Ankunftszeit beim Kunden
timeframe_start	Zeitfenster für die Ankunft Anfang
timeframe_end	Zeitfenster für die Ankunft Ende

Zeitplanung mit Zeitfenster Fehler 2:

Zwischen 2 Punkten reicht die Zeit nicht, sie innerhalb einer Tour anzufahren.

```
"time_window_error": {
  "type": 2,
  "nr1": 2,
  "nr2": 3,
  "name1": "Name Punkt1",
  "name2": "Name Punkt2",
  "timeframe_start1": "3:30 Uhr",
  "timeframe_end1": "3:45 Uhr",
  "timeframe_start2": "3:30 Uhr",
  "timeframe_end2": "3:45 Uhr",
  "start_kunde1": "4:30 Uhr",
  "stop_kunde2": "6:12 Uhr",
  "start_kunde2": "4:30 Uhr",
  "stop_kunde1": "6:10 Uhr",
  "fahrzeit12": "1h:42m",
  "fahrzeit21": "1h:40m",
  "aufenthalt_kunde1": "1h:00m",
  "aufenthalt_kunde2": "1h:00m"
}
```

Parameter	Bedeutung
time_window_error	
type	Fehlertyp 2: Zwischen 2 Kunden gibt es keine Möglichkeit, sie innerhalb einer Tour anzufahren.
nr1	Aktuelle Position Kunde 1 in den waypoints (Array beginnt mit 1)
nr2	Aktuelle Position Kunde 2 in den waypoints (Array beginnt mit 1)
name1	CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist
name2	CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist
timeframe_start1	Zeitfenster für die Ankunft Anfang (Kunde 1)
timeframe_end1	Zeitfenster für die Ankunft Ende (Kunde 1)
timeframe_start2	Zeitfenster für die Ankunft Anfang (Kunde 2)
timeframe_end2	Zeitfenster für die Ankunft Ende (Kunde 2)
start_kunde1	Abfahrtszeit bei Kunde 1 (timeframe_start1 + aufenthalt_kunde1)
stop_kunde2	Ankunftszeit bei Kunde 2 (start_kunde1 + fahrzeit12)
start_kunde2	Abfahrtszeit beim Depot (timeframe_start2 + aufenthalt_kunde2)
stop_kunde1	Abfahrtszeit beim Depot (start_kunde2 + fahrzeit21)
fahrzeit12	Fahrzeit von Kunde 1 zu Kunde 2
fahrzeit21	Fahrzeit von Kunde 2 zu Kunde 1
aufenthalt_kunde1	Aufenthaltsdauer bei Kunde 1
aufenthalt_kunde2	Aufenthaltsdauer bei Kunde 2

5 Webservice-Optimierung

5.1 Webservice-Optimierung-Aufruf

Wenn man die aktuelle Tour im Modus 'nur anzeigen' gestartet hat, kann man mit diesem Aufruf auch nachträglich die Optimierung starten

<https://webservice.tourenplaner.biz/optimize?tid=<transaktions-ID>>

oder

<https://webservice.tourenplaner.biz/optimize.json?tid=<transaktions-ID>>

5.2 GET Parameter

Zusätzlich kann man noch für die Zeitplanung mit Zeitfenster diese beiden Parameter angeben (als GET Parameter):

Parameter	Beschreibung	Werte
time_window	Optional Zeitplanung mit Zeitfenstern	true or false (default)
time_window_caclulation_time	Optional Maximale Zeit, die für die Berechnung der Zeitplanung mit Zeitfenstern angewendet wird. Funktioniert nur bei time_window: true	1-600 (60) default

Beispiel

```
curl -u test:test -H "Content-Type: application/json" "https://webservice.tourenplaner.biz/optimize?tid=123456"
curl -u test:test -H "Content-Type: application/json" "https://webservice.tourenplaner.biz/optimize?tid=123456&time_window=true&time_window_caclulation_time=120"
```

6 Postleitzahlen entlang der Tour

Um zusätzlich mögliche Kundenbesuche entlang einer Tour identifizieren zu können, besteht die Möglichkeit, nach den von der Tour tangierten PLZ-Gebieten zu suchen.

Mit den Ergebnissen kann z.B. im CRM, ERP, usw. nach weiteren Kunden gesucht werden („welche Kunden sind in den PLZ, die von der Tour berührt/geschnitten werden?“), die anschließend in die Tour integriert werden könnten.

6.1 Webservice Aufruf

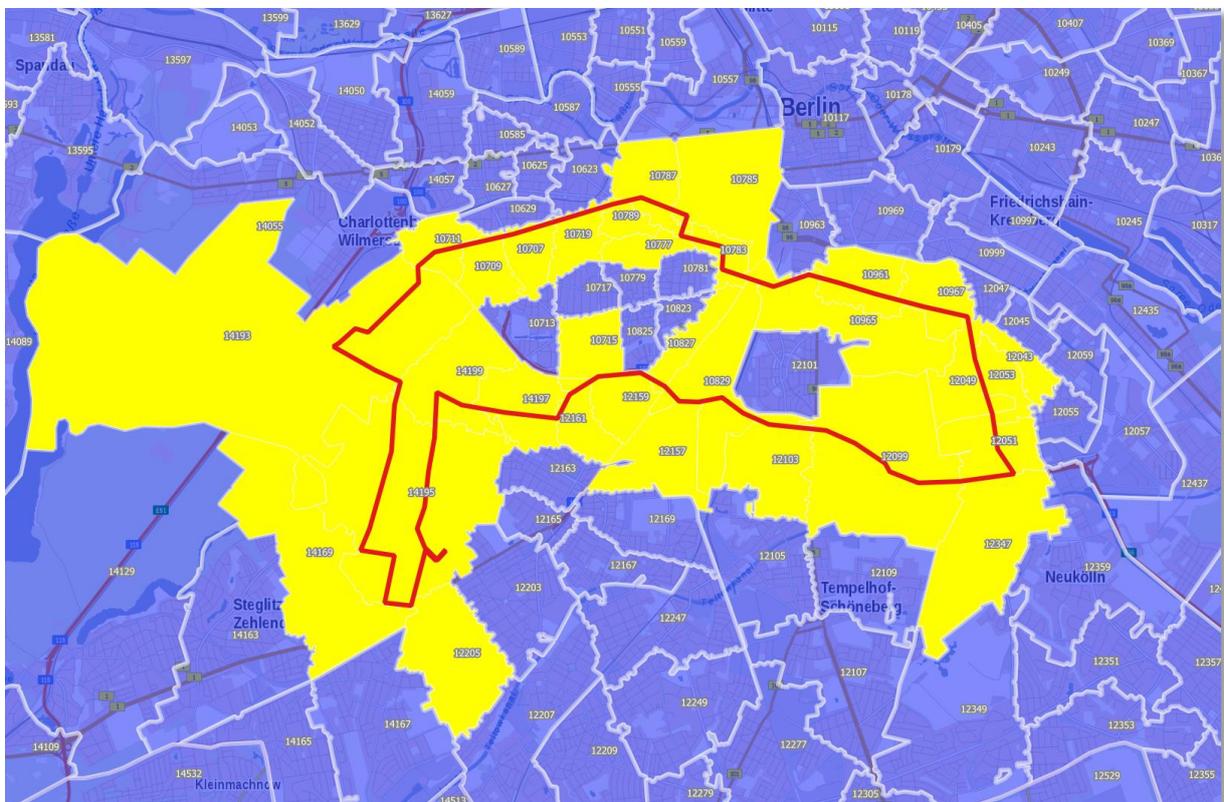
6.1.1 Ohne Puffer

https://webservice.tourenplaner.biz/get_postcodes_along_tour?tid=<transaktions-ID>

oder

https://webservice.tourenplaner.biz/get_postcodes_along_tour.json?tid=<transaktions-ID>

Dieser Aufruf liefert die Postleitzahlen aller PLZ-Gebiete zurück, die von der Tour-Linie berührt/geschnitten werden (im Bild unten gelb dargestellt).



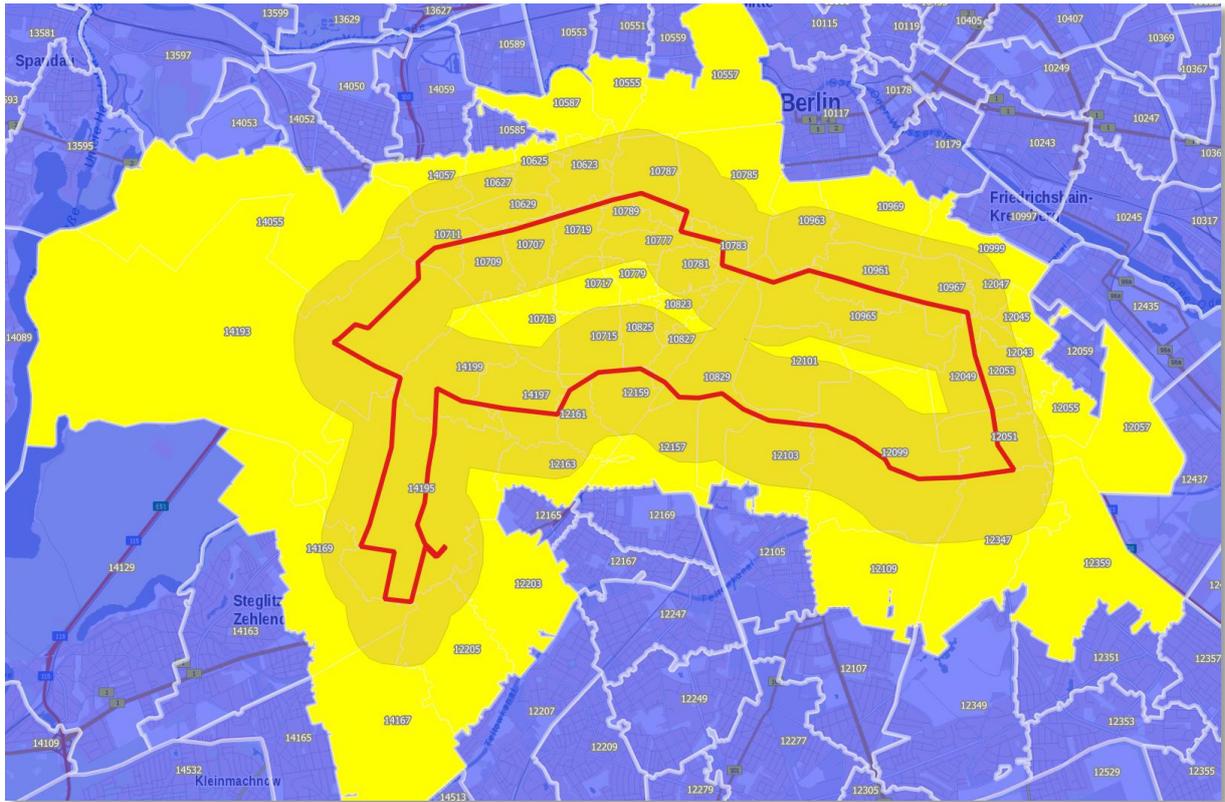
6.1.2 Optional zusätzlich mit Puffer

https://webservice.tourenplaner.biz/get_postcodes_along_tour?tid=<transaktions-ID>&buffer=<Meter>

oder

https://webservice.tourenplaner.biz/get_postcodes_along_tour.json?tid=<transaktions-ID>&buffer=<Meter>

Dieser Aufruf liefert die Postleitzahlen aller PLZ-Gebiete zurück, die von der der gepufferten-Tour-Linie (=Polygon) berührt/geschnitten werden (im Bild unten gelb dargestellt).



6.2 Ergebnis

```
{
  "postcodes": ["80796", "80797", "80801", "80802", "80803", "80805"]
}
```

6.3 Erweiterung

```
{
  "data": [
    {
      "country": "DE",
      "postcodes": ["80796", "80797", "80801", "80802", "80803", "80805"]
    },
    {
      "country": "AT",
      "postcodes": ["1000", "1001"]
    }
  ]
}
```

6.4 Voraussetzung

Für diese Funktion ist für Abfragen außerhalb Deutschlands der Erwerb von Lizenzen für PLZ Geometriedaten erforderlich (deutsche PLZ sind kostenlos enthalten).

7 Zusätzliche Punkte hinzufügen

Es können jederzeit zu einer Tour zusätzliche Punkte hinzugefügt werden.

7.1 “waypoints”

Geliefert werden muss ein Array mit dem einzelnen Objekt

Parameter	Beschreibung	Werte
uid	Optional CRM-uniqueID , unter welchem der Wegpunkt gespeichert ist	String
address	Required Adresse	Object
coordinate	Optional Koordinaten des Wegpunktes, keine Geokodierung notwendig	Object
time	Optional Zeitangabe zu dem Wegpunkt*	Object
name	Optional Name der angezeigt werden soll	String

Es können beliebige key value (String) Paare angegeben werden, diese werden einfach durchgereicht.

Parameter address

Parameter	Beschreibung	Werte
locality	Required Ort	Bsp.: München
postcode	Required Postleitzahl	Bsp.: 80331
street	Required Straße Hausnummer	Bsp.: Tal 1

Parameter coordinate

Parameter	Beschreibung	Werte
lat	Required Latitude	Bsp.: 48.5
lon	Required Longitude	Bsp.: 11.5

Parameter time

Parameter	Beschreibung	Werte
duration_of_stay	Required Aufenthaltszeit in Minuten	Bsp.: 60
arrival	Required Ankunftszeit	Bsp.: 2013-20-19T15:22:46+01:00
departure	Required Abhartszeit	Bsp.: 2013-20-19T15:22:46+01:00

Achtung

1. Zeitangaben in ISO 8601 Format
2. duration_of_stay wird nicht für den Ersten Punkt ausgewertet.
3. arrival, departure es wir nur ein fixer Termin verwendet.
 - a. Es wird beim ersten Datensatz wo es eine arrival oder departure Angabe findet als Fixtermin festgelegt.
 - b. Wenn arrival und departure angegeben sind, dann wird arrival als Fixtermin festgelegt.

7.2 Beispiel

```
[
  {
    "uid": "1",
    "coordinate" : {"lat": "48.163415", "lon": "11.577531"},
    "address": {"locality": "München", "postcode": "80803", "street":
"Pündterplatz 8"}
  },
  {
    "uid": "2",
    "coordinate" : {"lat": "49.454345", "lon": "11.073495"},
    "address": {"locality": "Nürnberg", "postcode": "90403", "street":
"Weintraubengasse 1"}
    "time": {"duration_of_stay" : 60}
  },
  {
    "uid": "3",
    "coordinate" : {"lat": "48.136415", "lon": "11.577531"},
    "address": {"locality": "München", "postcode": "80331", "street":
"Tal 1"}
  }
]
```

7.3 Curl Beispiel

```
curl -u test:test -XPOST -H "Content-Type: application/json" "https://web-
service.tourenplaner.biz/add_corridor_points?tid=AG-nhaaIPX6DkVwQCOMdlw" -d
'
{
  "waypoints": [
    {
      "uid": "20",
      "name" : "Test1",
      "address": {"locality": "Oberschleißheim", "postcode": "",
"street": ""}
    },
    {
      "uid": "21",
      "name" : "Test2",
      "address": {"locality": "Moosinning", "postcode": "85452",
"street": "Am Bleibach 4a"}
    },
    {
      "uid": "22",
      "name" : "Test3",
      "address": {"locality": "Ahrensburg", "postcode": "22926",
"street": "Beimoorkamp 6"}
    },
    {
      "uid": "23",
      "name" : "Test4",
      "address": {"locality": "Saarbrücken", "postcode": "66125",
"street": "Beim Weisenstein 13"}
    }
  ]
}'
```

8 Webservice-Download

https://webservice.tourenplaner.biz/download?tid=<transaktions-ID>&download_format=<format>

format	Name
alanwaypointsandrouteformat	Alan Map 500 Waypoints and Routes (*.wpr)
columbusv900professionalformat	Columbus V900 Professional (*.csv)
columbusv900standardformat	Columbus V900 Standard (*.csv)
compegpsdatarouteformat	CompeGPS Data Route (*.rte)
compegpsdatawaypointformat	CompeGPS Data Waypoint (*.wpt)
copilot6format	CoPilot 6 (*.trp)
copilot7format	CoPilot 7 (*.trp)
copilot8format	CoPilot 8 (*.trp)
copilot9format	CoPilot 9 (*.trp)
excel	Excel as of Version 2007 (.xlsx)
excel_old	Excel (.xls)
flightrecorderdataformat	FAI/IGC Flight Recorder Data (*.igc)
tourformat	Falk Navigator (*.tour)
garminflightplanformat	Garmin Flight Plan (*.fpl)
garminmapsource5format	Garmin MapSource 5.x (*.mps)
garminmapsource6format	Garmin MapSource 6.x (*.gdb)
garminpcx5format	Garmin PCX5 (*.wpt)
garminpoiformat	Garmin POI (*.gpi)
garminpoidbformat	Garmin POI Database (*.xcsv)
geocachingformat	Geocaching.com/EasyGPS (*.loc)
glopusformat	Glopus (*.tk)
kml20format	Google Earth 3 (*.kml)
kmz20format	Google Earth 3 Compressed (*.kmz)
kml21format	Google Earth 4 (*.kml)
kmz21format	Google Earth 4 Compressed (*.kmz)
kml22betaformat	Google Earth 4.2 (*.kml)
kmz22betaformat	Google Earth 4.2 Compressed (*.kmz)
kml22format	Google Earth 5 (*.kml)
kmz22format	Google Earth 5 Compressed (*.kmz)
gopal3routeformat	GoPal 3 Route (*.xml)
gopal5routeformat	GoPal 5 Route (*.xml)
goridergpsformat	GoRider GPS (*.rt)
gpx10format	GPS Exchange Format 1.0 (*.gpx)
gpx11format	GPS Exchange Format 1.1 (*.gpx)
gpstunerformat	GPS Tuner (*.trk)
haicomloggerformat	Haicom Logger (*.csv)
ibblue747format	i-Blue 747 (*.csv)
igo8routeformat	iGO8 Route (*.kml)
klicktelrouteformat	klickTel Routenplaner 2009 (*.krt)
kompasformat	Kompass (*.tk)
magellanexploristformat	Magellan Explorist (*.log)
magellanmapsendformat	Magellan MapSend (*.wpt)

magellanrouteformat	Magellan Route (*.rte)
magicmapsiktformat	MagicMaps Project (*.ikt)
magicmapspthformat	MagicMaps Tour (*.pth)
magicmaps2goformat	MagicMaps2Go (*.txt)
mtp0607format	Map&Guide Tourenplaner 2006/2007 (*.bcr)
mtp0809format	Map&Guide Tourenplaner 2008/2009 (*.bcr)
navigatingpoiwarnerformat	Navigating POI-Warner (*.asc)
nmnrouteformat	Navigon Mobile Navigator (*.route)
nmn4format	Navigon Mobile Navigator 4 (*.rte)
nmn5format	Navigon Mobile Navigator 5 (*.rte)
nmn6format	Navigon Mobile Navigator 6 (*.rte)
nmn6favoritesformat	Navigon Mobile Navigator 6 Favorites
nmn7format	Navigon Mobile Navigator 7 (*.freshroute)
nmnurlformat	Navigon Mobile Navigator URL (*.txt)
nmeaformat	NMEA 0183 Sentences (*.nmea)
nokialandmarkexchangeformat	Nokia Landmark Exchange (*.lmx)
opelnavigformat	Opel Navi 600/900 (*.poi)
oziexplorerouteformat	OziExplorer Route (*.rte)
oziexplorerwaypointformat	OziExplorer Waypoint (*.wpt)
qstarzq1000format	Qstarz BT-Q1000 (*.csv)
route66format	Route 66 POI (*.csv)
sygicunicodeformat	Sygyic POI Unicode (*.txt)
tomtom5routeformat	Tom Tom 5 Route (*.itn)
tomtom8routeformat	Tom Tom 8 Route (*.itn)
tomtompoiformat	Tom Tom POI (*.ov2)
ovlformat	Top50 OVL ASCII (*.ovl)
tcx1format	Training Center Database 1 (*.tcx)
tcx2format	Training Center Database 2 (*.tcx)
viamichelinformat	ViaMichelin (*.xvm)
webpageformat	Web Page (*.html)

9 Weitere Aufrufe

9.1 Tour löschen

<https://webservice.tourenplaner.biz/delete.json?tid=<tid>>

Rückgabe

HTTP-Statuscode 200 bzw. 500

9.2 Dynamische Karte

9.2.1 Karte mit einer Tour anzeigen

https://webservice.tourenplaner.biz/map_only/<tid>

Rückgabe

HTTP-Statuscode 200 bzw. 500

Die dynamische Karte kann z.B. als iFrame eingebunden werden.

9.2.2 Gleichzeitiges Anzeigen mehrerer Touren und eines Punktes

Diese Funktion kann genutzt werden, um für eine neue Adresse visuell zu bewerten, zu welcher bestehenden Tour diese am nächsten ist.

https://webservice.tourenplaner.biz/map_tours/<tids>?locality=<ort>&street=<straße>

Parameter	Kommentar	
tids	required	als Komma separierte Liste <tid>,<tid>...
locality	optional	Ort (Es wird geokodiert)
postcode	optional	Postleitzahl (Es wird geokodiert)
street	optional	Straße Hausnummer (Es wird geokodiert)
latitude	optional	Breitengrad (Es wird nicht geokodiert)
longitude	optional	Längengrad (Es wird nicht geokodiert)