



Bundesamt für  
Kartographie und Geodäsie

# Routingdienste des BKG

---

## Schnittstellenbeschreibung (Bund)

Version ORS 7.2.1

07.02.2024

Dienstleistungszentrum  
Bundesamt für Kartographie und Geodäsie

## Inhalt

Abkürzungsverzeichnis.....	4
Überblick über den Dienst.....	5
1 Verwendete Software, Daten und Lizenzen.....	7
2 Datenmodell.....	8
2.1 Projektion.....	8
2.2 Ortsangaben.....	8
2.2.1 Länge (lon).....	8
2.2.2 Breite (lat).....	8
3 Funktionen des Routingdienstes.....	9
3.1 Routingserver-API.....	9
3.2 Implementierte Dienste (Services) und Funktionen.....	14
3.3 Service viaroute (Standardroute).....	14
3.3.1 Anfrage/ Routing-Optionen (POST-Request).....	14
3.3.2 Antwort.....	19
3.3.3 Beispiel.....	21
3.3.4 Routing-Restriktionen.....	23
3.4 Service viaroute (Standardroute) mit statistischen Verkehrsfluss- informationen.....	23
3.5 Erreichbarkeitsanalyse.....	24
3.5.1 Anfrage.....	24
3.5.2 Antwort.....	26
3.5.3 Erreichbarkeitsanalysen-Restriktionen.....	28
3.6 Erreichbarkeitsanalyse mit statistischen Verkehrsfluss- informationen.....	28
3.7 Matrix.....	29
3.7.1 Anfrage.....	30
3.7.2 Antwort.....	31
3.7.3 Matrix-Restriktionen.....	32
3.8 Travel Salesman Routing.....	32
3.8.1 Anfrage.....	32
3.8.2 Antwort.....	33
3.8.3 Travel Salesman Restriktionen.....	34
4 Datenschutz.....	35
5 Quellenverzeichnis.....	36
6 Kontakt.....	37
6.1 Freischaltung.....	37

6.2 Technische Anfragen..... 37

## Abkürzungsverzeichnis

EPSG	European Petroleum Survey Group
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
geoJSON	JavaScript Object Notation für geographische Objekte
gpx	JavaScript Object Notation GPS Exchange Format
ORS	OpenRouteService
VROOM	Vehicle Routing Open-source Optimization Machine

## Überblick über den Dienst

Der Routingdienst für die Ermittlung von Routen und Durchführung von Erreichbarkeitsanalysen ist ein Projekt des Bundesamtes für Kartographie und Geodäsie (BKG) für die Bundesverwaltung. Als Datengrundlage für die Routenberechnung werden die frei verfügbaren Daten der Nutzergemeinschaft OpenStreetMap verwendet. Das BKG bereitet die Daten auf und betreibt die daraus abgeleiteten Webdienste.

Das vorliegende Dokument beschreibt die Schnittstellen des Routingdienstes und gibt Hinweise zu dessen Anwendung.

Ein *Routingdienst* ist ein Webdienst, der Wegberechnungen (Routen) zwischen mit Koordinaten versehenen Objekten unter Berücksichtigung von Geschwindigkeitsprofilen umsetzt und für eine Weiterverarbeitung zugänglich macht.

Der Routingdienst liegt in der Version Openrouteservice (ORS) 7.2.1 vor.

Die Ansprache des Routingdienstes ORS v. 7.2.1 erfolgt über [https://sg.geodatenzentrum.de/v2/web\\_ors?](https://sg.geodatenzentrum.de/v2/web_ors?)

Die Ansprache des ORS-Dienstes ist über die Ansprachemethode POST – Resquest möglich.

Die vorliegende Serverimplementierung stellt die folgenden Diensttypen bereit:

- *Route* (directions)  
Der Routingdienst unterstützt die Ermittlung einer schnellsten und einer kürzesten Route für verschiedene Geschwindigkeitsprofile einschließlich der Ausgabe von Routinganweisungen.  
Die Verwendung von Wegepunkten (Via-Points) und Sperrflächen (avoid\_polygons) ist optional möglich
- Umkreisanalysen (isochrones) auf Basis von Zeit- oder Entfernungsangaben  
Zu einer definierten Ortsangabe wird eine Umkreisanalyse (Erreichbarkeitsanalyse) in Abhängigkeit von einer vorgegebenen Zeit oder vorgegebenen Entfernung eine Fläche rund um die Ortsangabe berechnet, welche die Erreichbarkeit darstellt.
- Matrix, Routing zu einer Vielzahl von Punkten werden zwischen allen eingegebenen Punkten die Strecke und Fahrzeit (Routing) berechnet.
- Travel Salesman Routing, routet zu einer Vielzahl von Zielen in der günstigsten Reihenfolge (fastest/ schnellste Route).  
Start- und Endpunkt können identisch sein (Ring).

Es findet keine persistente Speicherung von Daten statt.

Datengrundlage des Dienstes sind die Datenbestände des OpenSource-Projektes „OpenStreetmap“, da zum heutigen Zeitpunkt existierende amtliche Geobasisdaten keine Routingdaten enthalten und somit nicht routingfähig sind.

Der Routingdienst unterstützt die Schnittstelle json, geojson oder gpx und wird über die folgende URL bereitgestellt

[https://sg.geodatenzentrum.de/web\\_ors/v2/\[profile\]?](https://sg.geodatenzentrum.de/web_ors/v2/[profile]?)

Im vorliegenden Dokument werden die Nutzung der Schnittstelle und das Ausgabeformat beschrieben.

## 1 Verwendete Software, Daten und Lizenzen

In der bereitgestellten Implementierung wird als Routing-Engine das Softwarepaket „OpenRouteService (ORS) v. 7.2.1“ unter Anwendung und Beachtung ihrer jeweiligen Nutzungsbedingungen verwendet und zur Nutzung bereitgestellt, sie unterliegt der Lizenz GNU GPL, V3

Die Berechnung der Routeninformation erfolgt auf der Basis von OpenStreetMap-Daten, welche regelmäßig von OpenStreetMap im Format pbf<sup>1</sup> weltweit geladen werden. Deren Nutzung erfolgt unter Beachtung und Anwendung der Lizenz „Open Data Commons Open Database License (ODbL)“

Für die Berechnung des Travel Salesman Routing wird die OpenSource Software Vehicle Routing Open-source Optimization Machine (VROOM) verwendet. Die Software VROOM verwendet für die Routenberechnung den OpenRouteService. Die Software VROOM unterliegt der BSD 2-Clause "Simplified" License.

---

<sup>1</sup> [http://wiki.openstreetmap.org/wiki/DE:PBF\\_Format](http://wiki.openstreetmap.org/wiki/DE:PBF_Format)

## 2 Datenmodell

Für die Berechnung der Routingfunktionen werden lediglich Koordinatenangaben benötigt.

### 2.1 Projektion

Derzeit wird nur die Projektion „WGS84 geographisch“ unterstützt.

### 2.2 Ortsangaben

#### 2.2.1 Länge (lon)

Die Angabe der geographischen Länge erfolgt in Dezimalschreibweise wie folgt

8.500038146972656

#### 2.2.2 Breite (lat)

Die Angabe der geographischen Breite erfolgt in Dezimalschreibweise wie folgt

50.19991683959961



## 3 Funktionen des Routingdienstes

### 3.1 Routingserver-API

Die Ansprache des ORS-Dienstes ist über die Ansprachemethode POST - Resquest möglich , die Abfrage ist nachfolgend beschrieben:

#### Url des Routingserver

```
server-address = sg.geodatenzentrum.de/web_ors/v2/directions
```

#### Routing: Geschwindigkeitsprofile

Auto	profile = {driving-car}
Schwerlastverkehr	profile = {driving-hgv}
Fahrrad	profile = {cycling-regular}
Fußgänger	profile = {foot-walking}
Rollstuhl	profile = {wheelchair}

#### POST-Request

```
Routenberechnung zwischen Start- und Endpunkt, Via-points (optional)
service = {directions}
profile= [driving-car/ driving-hgv/ cycling-regular/ foot-walking/
wheelchair]
Format= [json / geojson/ gpx]
```

#### Parameter:

**coordinates/ options/ instructions etc.** [Details s. 3.3.1](#)

#### Beispiele:

##### Einfache Routenberechnung:

```
curl -X POST \
'https://sg.geodatenzentrum.de/web_ors/v2/directions/driving-car/geojson' \
-H 'Content-Type: application/json; charset=utf-8' \
-H 'Accept: application/json, application/geo+json, application/gpx+xml,
img/png; charset=utf-8' \
-d '{"coordinates":[[8.83098,49.89201],[8.92367,49.8884]]}'
```

##### Routenberechnung mit Sperrfläche:

```
curl -X POST \
'https://sg.geodatenzentrum.de/web_ors/v2/directions/driving-car/geojson' \
-H 'Content-Type: application/json; charset=utf-8' \
-H 'Accept: application/json, application/geo+json, application/gpx+xml,
img/png; charset=utf-8' \
-d '{"coordinates":[[8.8368946,49.8949312],[8.6820917,50.1106444]],"options":{"avoid_polygons":{"type":"Polygon","coordinates":[[[8.867296494174184,50.084046873456664],[8.931154526400734,50.08316567503269],[8.924288071322625,50.04261303450556],[8.852876938510109,50.03423376128765],[8.867296494174184,50.084046873456664]],[[8.680528916049164,50.
```

```
03423376128765],[8.743013657260093,50.035997940337225],[8.754686630892925,49.97156340612763],[8.667482651400713,49.975096302291625],[8.680528916049164,50.03423376128765]]]]}}}'
```

### geojson-Response

<b>FeatureCollection</b>	features
<b>bbox</b>	Bounding-Box der Route
<b>segments</b>	Routensegmente mit Fahrhinweisen
<b>summary</b>	distance [m], duration [Sek.]
<b>geometry</b>	coordinates": [[xx,yy],[xx,yy]...]
<b>way-points</b>	Wegepunkte, falls vorhanden
<b>info</b>	Informationen zur Routingserverversion, sowie zu den Requestparametern

### Fehler

Diese Rückgabe erfolgt, falls die Route nicht berechnet werden konnte

```
{"error":{"code":number,"message":"text"}}
```

Error Code	Description
2000	Unable to parse JSON request.
2001	Required parameter is missing.
2002	Invalid parameter format.
2003	Invalid parameter value.
2004	Parameter value exceeds the maximum allowed limit.
2006	Unable to parse the request to the export handler.
2007	Unsupported export format.
2008	Empty Element.
2009	Route could not be found between locations.
2099	Unknown internal error.

### Erreichbarkeitsanalysen:

#### Isochronen

```
Service= {isochrones} range_type=time
```

#### Distanzen

```
Service= {isochrones} range_type=distance
```

### POST-Request

---

Erreichbarkeitsanalysen ausgehend vom Startpunkt, Via-points (service = {isochrones} range\_type time/ distance [Details s. 3.4.1](#))

### Beispiele:

#### Erreichbarkeitsanalysen:

##### Isochronen

```
curl -X POST \
  'https://sg.geodatenzentrum.de/web_ors/v2/isochrones/driving-car/geojson' \
  -H 'Content-Type: application/json; charset=utf-8' \
  -H 'Accept: application/json, application/geo+json, application/gpx+xml, img/png; charset=utf-8' \
  -d '{"locations": [[8.681495, 49.41461], [8.686507, 49.41943]], "range": [30, 60], "range_type": "time"}'
```

##### Distanzen

```
curl -X POST \
  'https://sg.geodatenzentrum.de/web_ors/v2/isochrones/driving-car/geojson' \
  -H 'Content-Type: application/json; charset=utf-8' \
  -H 'Accept: application/json, application/geo+json, application/gpx+xml, img/png; charset=utf-8' \
  -d '{"locations": [[8.681495, 49.41461], [8.686507, 49.41943]], "range": [15000, 30000], "range_type": "distance"}'
```

### geojson-Response

<b>FeatureCollection</b>	features
<b>center</b>	Startpunkt des jeweiligen Umrings
<b>geometry</b>	Polygon-Koordinaten der Umringe der einzelnen Ranges coordinates: [[xx,yy], [xx,yy]...]
<b>bbox</b>	Bounding-Box der Route
<b>metadata</b>	Informationen zur Routingserverversion, sowie zu den Requestparametern

### Fehler

Diese Rückgabe erfolgt, falls die Route nicht berechnet werden konnte

```
{"error":{"code":number,"message":"text"}}
```

Error Code	Description
3000	Unable to parse JSON request.
3001	Required parameter is missing.
3002	Invalid parameter format.
3003	Invalid parameter value.

```
3004 Parameter value exceeds the maximum allowed limit.
3005 Requested feature is not supported.
3006 Unable to parse the request to the export handler.
3007 Unsupported export format.
3008 Empty Element.
3099 Unknown internal error.
```

### matrix: POST-Request

Erstellung von Eins-zu-Viel-, Viel-zu-Eins- und Viel-zu-Viel-Matrizen für Zeit und Entfernung [Details s. 3.5.1](#)

#### Beispiel:

```
curl -X POST \
  'https://sg.geodatenzentrum.de/web_ors/v2/matrix/driving-car' \
  -H 'Content-Type: application/json; charset=utf-8' \
  -H 'Accept: application/json, application/geo+json, application/gpx+xml,
img/png; charset=utf-8' \
  -d '{"locations":[[[9.70093,48.477473],[9.207916,49.153868],[37.573242,55.801281],[115
.663757,38.106467]]]}'
```

### geojson-Response

Parameter	Beschreibung
<b>distances/ durations</b>	je nachdem welcher Wert in "Metrics" gewählt wurde
<b>distances oder du- rations</b>	[mm, mm...] oder (sek,sek...]
<b>destinations</b>	Koordinaten der Zielpunkte[[xx,yy],[xx,yy]...]
<b>metadata</b>	Informationen zur Routingserverversion, sowie zu den Requestparametern

### Fehler

Diese Rückgabe erfolgt, falls die Route nicht berechnet werden konnte

```
{"error":{"code":number,"message":"text"}}
```

```
Error Code    Description
6000 Unable to parse JSON request.
6001 Required parameter is missing.
6002 Invalid parameter format.
6003 Invalid parameter value.
6004 Parameter value exceeds the maximum allowed limit.
6006 Unable to parse the request to the export handler.
6007 Unsupported export format.
6008 Empty Element.
6099 Unknown internal error.
```

### Travel Salesman Routing

#### POST-Request

Erstellung der günstigsten (schnellsten) Routenkombination zum Anfahren mehrerer Ziele

#### Beispiel:

```
curl -X POST 'https://sg.geodatenzentrum.de/web_ors/tsr/' --header "Content-Type: application/json" --data '{"vehicles":[{"id":0,"profile":"driving-car","start":[8.6526,49.4604],"end":[8.6526,49.4604]}],"jobs":[{"id":0,"location":[8.6691,49.4532]},{"id":1,"location":[8.6911,49.4566]}],"options":{"g":true}}'
```

#### json-Response

Parameter	Beschreibung
<b>code</b>	status code 0= kein Error
<b>summary</b>	Informationen Gesamtroute
<b>distances oder durations</b>	[mm, mm...] oder (sek,sek...]
<b>routes</b>	Informationen Einzelroute
<b>steps</b>	Koordinaten der Routenpunkte start[lon/lat]... job[lon/lat] (Ziele) ... end[lon/lat]
<b>geometry</b>	Polylinienkodierte Routengeometrie (pro step)

#### Fehler

Diese Rückgabe erfolgt, falls die Route nicht berechnet werden konnte

```
{"error":{"code":number,"message":"text"}}
```

Error Code	Description
6000	Unable to parse JSON request.
6001	Required parameter is missing.
6002	Invalid parameter format.
6003	Invalid parameter value.
6004	Parameter value exceeds the maximum allowed limit.
6006	Unable to parse the request to the export handler.
6007	Unsupported export format.
6008	Empty Element.
6099	Unknown internal error.

## 3.2 Implementierte Dienste (Services) und Funktionen

In der vorliegenden Implementierung sind folgende Dienste und Funktionen enthalten:

Service (Dienst)	Beschreibung
<b>directions</b>	schnellste oder kürzeste Route zwischen zwei oder mehr mit Koordinaten beschriebenen Punkten für Auto/ Fahrrad/ Fußgänger/ Schwerlastverkehr/ Rollstuhl
<b>isochrones</b>	Erreichbarkeitsanalyse, Isochronen/ Distanzvorgabe für Auto/ Fahrrad/ Fußgänger/ Schwerlastverkehr
<b>matrix</b>	Ermöglicht die Berechnung der Strecken und Fahrzeiten zwischen allen Punkten, die über den Parameter „locations“ übergeben werden. Ist eine Verbindung null konnte die betreffende Route nicht berechnet werden.
<b>Travel Salesman Routing</b>	Ermöglicht die Berechnung der günstigsten (schnellsten) Route zum Anfahren mehrer Ziele. Dabei werden die Zielpunkte zwischen dem Start- und Zielpunkt in der geeignetsten Reihenfolge ausgegeben. Für die Berechnung eines Rings müssen Star- und Endpunkt identisch sein.

## 3.3 Service viaroute (Standardroute)

Der Dienst ermittelt die schnellste oder kürzeste Route zwischen einem Start- und einem Zielpunkt sowie ggf. eingefügten Zwischenpunkten. Optional können die dazugehörigen Routinginstruktionen ausgegeben werden.

### 3.3.1 Anfrage/ Routing-Optionen (POST-Request)

```
curl -X POST \
  'https://sg.geodatenzentrum.de/web_ors/v2/directions/driving-car/geojson' \
  -H 'Content-Type: application/json; charset=utf-8' \
  -H 'Accept: application/json, application/geo+json, application/gpx+xml, img/png; charset=utf-8' \
  -d '{"coordinates":[[[8.83098,49.89201],[8.92367,49.8884]]]}'
```

Parameter	Wert	Beschreibung
<b>profile</b>	driving-car, driving-hgv, cycling-regular, foot-walking, wheelchair	Geschwindigkeitsprofile Auto, Schwerlastverkehr, Fahrrad, Fußgänger oder Rollstuhl

<b>coordinates</b>	lon, lat	Mit Koordinaten (Länge, Breite) beschrieben Punkte, die Koordinatenangaben erfolgen im System WGS84 in Dezimalschreibweise getrennt durch ein Komma (,)
<b>preference</b>	fastest, shortest, recommended	<p><b>fastest</b> richtet sich nach der Zeit, die Route wird anhand der Graphen so zusammengestellt, dass möglichst wenig Zeit von Start- zum Zielpunkt benötigt wird.</p> <p><b>shortest</b> richtet sich nach der Distanz, die Route wird anhand der Graphen so zusammengestellt, dass möglichst wenig Wegstrecke von Start- zum Zielpunkt benötigt wird.</p> <p><b>recommended</b> die empfohlene Routenpräferenz prüft zudem z.B. wie gut eine Route zum Radfahren geeignet ist. Dies geschieht dadurch, dass jeder Straße eine Priorität zugewiesen wird. Die Priorität wird aus den Tags berechnet, z. B. Radweg ob es Geschwindigkeitsminderungen aufgrund der Steigung gibt. Diese Priorität wird dann in Kombination mit der Option „Schnellste“ verwendet, um eine empfohlene Route auszugeben. Daher liefern Requests mit der Anfrage "shortest"/ "recommended" unterschiedliche Ergebnisse.</p>
<b>alternative_route</b>	share_factor, target_count, weight_factor	<p>Gibt an, ob Alternativrouten berechnet werden, sowie Parameter für den Algorithmus zur Ermittlung geeigneter Alternativen.</p> <p><b>share_factor</b> Maximaler Anteil der Route, den sich Alternativen mit der optimalen Route teilen dürfen. Der Standardwert von 0,6 bedeutet, dass die Alternativen bis zu 60 % der Pfadsegmente mit der optimalen Route teilen können.</p> <p><b>target_count</b> Zielanzahl der zu berechnenden Alternativrouten. Der Dienst gibt bis zu dieser Anzahl von Routen zurück, die die Bedingungen für den Anteilfaktor und den Gewichtungsfaktor erfüllen.</p> <p><b>weight_factor</b> Maximaler Faktor, um den die Routengewichtung von der optimalen Route abweichen darf. Der Standardwert von 1,4 bedeutet, dass die Alternativen bis zu 1,4 Mal länger (teurer) sein können als die optimale Route.</p>
<b>attributes</b>	avgspeed, percentage, dertourfactor	<p><b>avgspeed</b> Dieser Wert wird in km/h angegeben und entspricht der Durchschnittsgeschwindigkeit für diesen Streckenabschnitt</p> <p><b>percentage</b></p>

		<p>Dieser Wert wird in Prozent angegeben und gibt die Segmentlänge in Bezug auf die Streckenlänge an <b>dertourfactor</b></p> <p>Dieser Wert ist ein Faktor und gibt die relative Länge des Abschnitts im Verhältnis zur Länge der Luftlinie zwischen dem Anfangs- und Endpunkt des Streckenabschnitts an.</p>
<b>continue_straight</b>	true, false	Zwingt die Route, an Wegpunkten geradeaus zu fahren und dort nicht abzubiegen, auch wenn dies schneller wäre.
<b>elevation</b>	true, false	Gibt an, ob Höhenwerte für Punkte zurückgegeben werden sollen. Bitte beachten Sie, dass die Höhe auch für die in der json-Antwort kodierte Polylinie kodiert wird.
<b>extra_info</b>	stepness, suitability, surface, waycategory, waytype, tollways, traildifficulty, osmid, roadaccessrestrictions, countryinfo, green, noise	<p>Stufigkeit, Eignung, Oberfläche, Weg-Kategorie, Wegetyp, Mautstraßen, Verkehrstauglichkeit, osmid= Sonderparameter für Rollstuhlrouting Straßenzugangsbeschränkungen, Länderinformationen, Landschaftlich schöne Route Ruhige, unbefahrene Route</p>
<b>geometry_simplify</b>	true, false	Gibt an, ob die Geometrie vereinfacht werden soll. Geometrie vereinfachen kann nicht auf Strecken mit mehr als einem Segment angewendet werden und wenn extra_info verwendet wird
<b>id</b>	text	Beliebige Identifizierungszeichenfolge der Anfrage, die sich in den Metainformationen widerspiegelt
<b>instructions</b>	true, false	Gibt an, ob Routing-Anweisungen zurückgegeben werden sollen.
<b>language</b>	text	Sprache der Routing-Anweisungen
<b>options</b>	avoid_borders all/ controlled/ none	alle für keinen Grenzübergang kontrolliert, um offene Grenzen zu überschreiten, aber kontrollierte Grenzen zu vermeiden. Nur für driving- Profile.
	avoid_countries (array) [number]	Liste der Länder, die von der Streckenführung mit Fahrprofilen auszuschließen sind. Kann zusammen



	<p>z.B. [11,193]</p> <p>avoid_features (array) [text] Highway/ toll- ways/ ferries z.B. ["high- ways"]</p> <p>avoid_polygons geojson-Objekt</p> <p>round_trip (object) z.B. { "length":10000, "points":5}</p>	<p>mit 'avoid_borders' verwendet werden: 'controlled'. [ 11, 193 ] würde Österreich und die Schweiz ausschließen. Eine Liste von Ländern und Anwendungsbeispielen finden Sie <a href="#">hier</a>. Es können auch ISO-Standard-Ländercodes anstelle der numerischen IDs verwendet werden, z. B. DE oder DEU für Deutschland.</p> <p>Liste der zu vermeidenden Verkehrstypen.</p> <p>Umfasst die für die Route zu vermeidenden Gebiete. Formatiert in GeoJSON entweder als Polygon- oder Multipolygon-Objekt</p> <p>Optionen, die auf Hin- und Rückfahrtstrecken anzuwenden sind.</p>
<b>maneuvers</b>	<p>true, false</p>	<p>gibt an, ob Manöverobjekte verwendet werden</p>
<b>radiuses</b>	<p>(array) [number] z.B. [200,-1,30]</p>	<p>Eine Liste von maximalen Entfernungen (gemessen in Metern), die Suche nach nahe gelegenen Straßenabschnitten zu jedem gegebenen Wegpunkt einschränken. Die Werte müssen größer als 0 sein, der Wert -1 bedeutet keine Begrenzung der Suche. Die Anzahl der Radien entspricht der Anzahl der Wegpunkte. Wird nur ein einziger Wert angegeben, so wird dieser auf alle Wegpunkte angewandt.</p>
<b>roundabout_exits</b>	<p>true, false</p>	<p>Liefert die Peilungen der Einfahrt und aller passierten Ausfahrten des Kreisverkehrs. Fügt das Array exit_bearings dem Step-Objekt in der Antwort hinzu.</p>
<b>skip_segments</b>	<p>(array) [number] z.B. [2,4]</p>	<p>Gibt die Segmente an, die bei der Routenberechnung übersprungen werden sollen. Ein Segment ist die Verbindung zwischen zwei gegebenen Koordinaten,</p>

		die Zählung beginnt mit 1 für die Verbindung zwischen der ersten und der zweiten Koordinate.
<b>suppress_warnings</b>	true, false	Warnmeldungen in der Antwort unterdrücken
<b>units</b>	Text m/ km/ mi	Ausgabeinheit wählen Meter/ Kilometer/ Meilen
<b>geometry</b>	true, false	Ausgabe der Geometriedaten
<b>maximum_speed</b>	number	
		Die vom Benutzer angegebene Höchstgeschwindigkeit

Parameter für Schwerlastverkehr	Wert	Beschreibung
<b>options → profile_params</b>	length number	Länge in Meter
	width number	Breite in Meter
	Height number	Höhe in Meter
	axleload number	Achslast in Tonnen
	weight number	Gesamtgewicht in Tonnen
	hazmat true, false (default)	Gefahrguttransport

Parameter für Rollstuhlrou- ten	Wert	Beschreibung
<b>surface_type</b>	cobblestone:flattened (default), paved, unpaved, asphalt, sett, concrete, concrete:lanes, concrete:plates, paving_stones, cobblestone, metal, wood,	Beschaffenheit der Fahrbahn, die für die Routenberechnung genutzt werden kann  deatillierte Beschreibung unter: <a href="http://wiki.openstreet-map.org/wiki/Key:surface">http://wiki.openstreet-map.org/wiki/Key:surface</a>

	<code>compacted,</code> <code>dirt,</code> <code>earth,</code> <code>grass</code>	
<b>track_type</b>	<code>grade1 (default)</code> <code>.</code> <code>.</code> <code>.</code> <code>grade5</code>	Grad der Oberflächenfestigkeit  detaillierte Beschreibung unter: <a href="http://wiki.openstreetmap.org/wiki/Key:tracktype">http://wiki.openstreetmap.org/wiki/Key:tracktype</a>
<b>smoothness_type</b>	<code>good (default),</code> <code>excellent,</code> <code>intermediate,</code> <code>bad</code> <code>.</code> <code>.</code> <code>.</code> <code>impassable</code>	Glättungsgrad der Oberfläche  detaillierte Beschreibung unter: <a href="http://wiki.openstreetmap.org/wiki/Key:smoothness">http://wiki.openstreetmap.org/wiki/Key:smoothness</a>
<b>maximum_sloped_curb</b>	<code>number 0.06 (default)</code> <code>0.01 bis 0.99 [m]</code>	Maximale Höhe von Bordsteinkanten [m]
<b>maximum_incline</b>	<code>number 6 (default)</code> <code>0 bis 100 [%]</code>	Steigung in Prozent

### 3.3.2 Antwort

Parameter	Beschreibung
<b>FeatureCollection</b>	<code>features</code>
<b>features</b>	<p><code>bbox:</code>                      Bounding-Box des jeweiligen Features</p> <p><code>segments:</code>                      ein Array, welches die Routinganweisungen für jedes Routensegment in der nachfolgenden Form enthält. Jeder Eintrag hat die Array-Elemente:  <code>[{type Fahrt/Abbiegerichtung}, {instruction Fahranweisung}, {name Straßename}, {way_points Segmentnummer von...bis}, {distance Segmentlänge}, {duration Fahrzeit für das Segment}]</code></p> <ul style="list-style-type: none"> <li><code>type</code> Integer-Code der Abbiegerichtung (0 bis 11)</li> <li><code>instruction</code> Fahranweisung als string</li> </ul>

	<ul style="list-style-type: none"> <li>• name            Straßenname   als string</li> <li>• way_points    Start/ Endcode des Segments Integer</li> <li>• distance       Steckenlänge des Segments [m/ km oder mi] als Float</li> <li>• duration       Fahrzeit für das Segment als Float</li> </ul> <p>summary: Zusammenfassung der gelieferten Route</p> <ul style="list-style-type: none"> <li>• distance - Gesamtstrecke in Metern als integer</li> <li>• duration - Gesamtreisezeit in Sekunden als integer</li> </ul> <p>geometry: coordinates array [lon,lat]</p> <p>type: Objektyp z.B. LineString</p>
<b>bbox</b>	<p>Zusammenfassung der gelieferten Route</p> <ul style="list-style-type: none"> <li>• distance - Gesamtstrecke in Metern als integer</li> <li>• duration - Gesamtreisezeit in Sekunden als integer</li> </ul>
<b>metadata</b>	<p>attribution: Quellenangaben</p> <p>service. Servicetyp z.B. outing</p> <p>timestamp: Zeitstempel der Routinganfrage</p> <p>query: json-Object { profile: "driving-car", format: "geojson", coordinates: [...] }</p> <p>coordinates: Koordinaten des Start- u. Zielpunktes / evtl. Via-Points</p> <p>profile: Fahrzeugtyp driving-car/ driving-hgv/ cycling-regular/ foot-walking/ wheelchair</p> <p>format:        z.B. geojson</p>

	<pre>engine      Object: Angaben zur RoutingEngine, Version, Erstellungs- datum der Software, Erstellungsdatum der Graphen   { version:  "6.7.0",  build_date:  "2021-06- 08T13:11:04Z",      graph_date:    "2021-06- 16T12:52:16Z" } version     "6.7.0"  build_date: Erstellungsdatum der Software graph_date: Erstellungsdatum der Graphen</pre>
--	--

### 3.3.3 Beispiel

#### Beispiele:

##### **Einfache Routenberechnung z.B. schnellste Route:**

```
curl -X POST 'https://sg.geodatenzentrum.de/web_ors_all/v2/directions/driving-car/geojson' -H 'Content-Type: application/json; charset=utf-8' -H 'Accept: application/geo+json; charset=utf-8' -d '{"coordinates":[[[7.163841,51.198603],[7.312,51.297994]],{"preference":"fastest"}'
```

##### **Routenberechnung mit Sperrfläche:**

```
curl -X POST \
'https://sg.geodatenzentrum.de/web_ors/v2/directions/driving-car/geojson' \
-H 'Content-Type: application/json; charset=utf-8' \
-H 'Accept: application/json, application/geo+json, application/gpx+xml, img/png; charset=utf-8' \
-d '{"coordinates":[[[8.8368946,49.8949312],[8.6820917,50.1106444]],{"options":{"avoid_polygons":{"type":"Polygon","coordinates":[[[8.867296494174184,50.084046873456664],[8.931154526400734,50.08316567503269],[8.924288071322625,50.04261303450556],[8.852876938510109,50.03423376128765],[8.867296494174184,50.084046873456664]],[[8.680528916049164,50.03423376128765],[8.743013657260093,50.035997940337225],[8.754686630892925,49.97156340612763],[8.667482651400713,49.975096302291625],[8.680528916049164,50.03423376128765]]]]}}}'
```

Antwort:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "bbox": [8.831064, 49.880078, 8.923668, 49.892078],
      "type": "Feature",
      "properties": {
        "segments": [
          {
            "distance": 8080.5,
            "duration": 554.6,
            "steps": [
              {
                "distance": 91.3,
                "duration": 21.9,
                "type": 11,
                "instruction": "Head southeast on Feldstraße",
                "name": "Feldstraße",
                "way_points": [0, 5]
              },
              {
                "distance": 233.8,
                "duration": 30.6,
                "type": 1,
                "instruction": "Turn right onto Groß-Zimmerner Straße",
                "name": "Groß-Zimmerner Straße",
                "way_points": [5, 17]
              },
              {
                "distance": 121.1,
                "duration": 20.1,
                "type": 7,
                "instruction": "Enter the roundabout and take the 2nd exit onto Groß-Zimmerner Straße, L 3114",
                "name": "Groß-Zimmerner Straße, L 3114",
                "exit_number": 2,
                "way_points": [17, 30]
              },
              {
                "distance": 55.4,
                "duration": 2.2,
                "type": 13,
                "instruction": "Keep right onto B 26",
                "name": "B 26",
                "way_points": [30, 36]
              },
              {
                "distance": 39.6,
                "duration": 1.6,
                "type": 2,
                "instruction": "Turn sharp left onto B 26",
                "name": "B 26",
                "way_points": [36, 39]
              },
              {
                "distance": 252.8,
                "duration": 38.2,
                "type": 1,
                "instruction": "Turn right onto L 3114",
                "name": "L 3114",
                "way_points": [39, 44]
              },
              {
                "distance": 2045.6,
                "duration": 66.1,
                "type": 12,
                "instruction": "Keep left onto B 26",
                "name": "B 26",
                "way_points": [44, 88]
              },
              {
                "distance": 1139.6,
                "duration": 66.9,
                "type": 13,
                "instruction": "Keep right onto B 26, B 45",
                "name": "B 26, B 45",
                "way_points": [88, 114]
              },
              {
                "distance": 2118.1,
                "duration": 137.2,
                "type": 6,
                "instruction": "Continue straight onto B 45",
                "name": "B 45",
                "way_points": [114, 140]
              },
              {
                "distance": 54.8,
                "duration": 12.3,
                "type": 6,
                "instruction": "Continue straight onto B 45",
                "name": "B 45",
                "way_points": [140, 144]
              },
              {
                "distance": 1395.4,
                "duration": 105.5,
                "type": 0,
                "instruction": "Turn left onto Semder Straße, L 3115",
                "name": "Semder Straße, L 3115",
                "way_points": [144, 165]
              },
              {
                "distance": 532.8,
                "duration": 52.1,
                "type": 7,
                "instruction": "Enter the roundabout and take the 2nd exit onto Semder Straße, L 3115",
                "name": "Semder Straße, L 3115",
                "exit_number": 2,
                "way_points": [165, 184]
              },
              {
                "distance": 0.0,
                "duration": 0.0,
                "type": 10,
                "instruction": "Arrive at Semder Straße, L 3115, on the right",
                "name": "-",
                "way_points": [184, 184]
              }
            ]
          }
        ],
        "summary": {
          "distance": 8080.5,
          "duration": 554.6,
          "way_points": [0, 184]
        },
        "geometry": {
          "coordinates": [
            [8.831113, 49.892078],
            [8.831162, 49.888088],
            [8.831064, 49.888066],
            [8.831188, 49.888055],
            [8.831367, 49.888051],
            [8.831614, 49.888064],
            [8.831721, 49.88736],
            [8.831771, 49.887043],
            [8.831792, 49.886912],
            [8.831842, 49.886572],
            [8.831963, 49.885801],
            [8.831988, 49.885746],
            [8.832055, 49.885724],
            [8.832301, 49.88573],
            [8.832424, 49.885748],
            [8.832445, 49.885751],
            [8.832597, 49.885768],
            [8.832799, 49.885787],
            [8.833045, 49.885799],
            [8.833314, 49.885826],
            [8.873261, 49.887821],
            [8.873554, 49.887766],
            [8.873738, 49.887732],
            [8.874018, 49.887677],
            [8.874931, 49.887497],
            [8.875784, 49.88738],
            [8.876789, 49.887275],
            [8.878859, 49.8871],
            [8.879416, 49.887028],
            [8.88007, 49.886925],
            [8.88074, 49.8868],
            [8.881425, 49.886647],
            [8.882032, 49.886485],
            [8.882635, 49.886309],
            [8.884397, 49.885694],
            [8.886088, 49.885085],
            [8.88639, 49.884977],
            [8.888811, 49.884122],
            [8.890674, 49.883483],
            [8.892367, 49.882891],
            [8.897104, 49.881259],
            [8.898135, 49.880908],
            [8.899174, 49.88055],
            [8.899519, 49.880431],
            [8.899759, 49.880345],
            [8.899868, 49.880306],
            [8.89995, 49.88028],
            [8.900233, 49.880183],
            [8.900378, 49.880136],
            [8.900547, 49.880078],
            [8.900668, 49.880168],
            [8.90081, 49.880272],
            [8.901072, 49.880481],
            [8.901213, 49.880547],
            [8.901404, 49.880619],
            [8.902192, 49.880936],
            [8.902667, 49.881128],
            [8.903916, 49.881618],
            [8.905018, 49.882057],
            [8.907036, 49.882874],
            [8.908082, 49.883298],
            [8.909018, 49.883677],
            [8.911061, 49.884503],
            [8.911995, 49.884881],
            [8.91302, 49.885293],
            [8.913131, 49.885334],
            [8.914452, 49.885824],
            [8.915165, 49.886087],
            [8.915671, 49.886279],
            [8.916345, 49.886553],
            [8.916984, 49.886755],
            [8.91702, 49.886749],
            [8.917057, 49.886749],
            [8.917097, 49.886757],
            [8.917133, 49.886771],
            [8.91716, 49.886791],
            [8.917176, 49.886812],
            [8.917183, 49.886836],
            [8.917256, 49.886879],
            [8.917695, 49.88711],
            [8.91992, 49.888032],
            [8.920359, 49.888193],
            [8.920892, 49.888312],
            [8.921207, 49.888345],
            [8.921561, 49.888354],
            [8.921582, 49.888354],
            [8.922047, 49.888371],
            [8.922529, 49.888387],
            [8.923618, 49.888425],
            [8.923668, 49.888425]
          ]
        }
      }
    ]
  }
}
```

```
88427]], "type": "Lin-  
eString"}]], "bbox": [8.831064, 49.880078, 8.923668, 49.892078], "metadata": {"at-  
tribution": "openrouteservice.org | OpenStreetMap contributors", "ser-  
vice": "routing", "timestamp": 1586246332418, "query": {"coordi-  
nates": [[8.83098, 49.89201], [8.92367, 49.8884]], "profile": "driving-car", "for-  
mat": "geojson"}, "engine": {"version": "6.0.0", "build_date": "2020-03-  
02T08:58:53Z", "graph_date": "2019-12-09T01:00:00Z"}}}
```

### 3.3.4 Routing-Restriktionen

- Routenlänge max. 6000 Km
- Sperrflächen für Routen bis max. 150 Km Länge, Größe der Sperrfläche max. 200 Km<sup>2</sup>
- maximale Anzahl Alternativrouten= 3
- maximale Routen-Wegepunkte= 50

## 3.4 Service viaroute (Standardroute) mit statistischen Verkehrsfluss- informationen

Für die Profile driving-car (Pkw) und driving-hvg (Schwerlastverkehr) besteht die Option die Routen mit statistischen Verkehrsflussinformationen zu berechnen. Hierbei fließen die statistisch verifizierten, zu erwartenden Verkehrsflüsse der letzten 2 Jahre in die Routenberechnung ein. Die statistischen Verkehrsflussinformationen liegen für Bundesautobahnen, Bundesstraßen, Landes- u. Kreisstraßen, sowie zum Teil für Gemeindestraßen vor. Die aktuell hinterlegten Verkehrsflussinformationen basieren auf Verkehrsdaten der Jahre 2018-2019. Der ORS-Dienst zur Routenberechnung mit stat. Verkehrsdaten liegt in der Version 7.2.0 vor.

Die Funktion des Routingdienstes mit statistischen Verkehrsflussinformationen wird mit dem Zusatz **\_st** in der URL angesprochen. (Dieser Dienst steht ausschließlich Bundesbehörden zur Verfügung)

### Url des Routingsservers mit statistischen Verkehrsflussinformationen

```
server-address = sg.geodatenzentrum.de/web_ors/v2/directions_st/[profil  
car/hgv]
```

Zur Routenberechnung mit statistischen Verkehrsflussinformationen werden im Request folgende zusätzliche Parameter angegeben:

- **location\_type**
  - **departure**  
Routing vom Startpunkt ausgehend (Fahrzeit mit stat. Verkehrsdaten)
  - **arrival**

Routing vom Startpunkt ausgehend unter Angabe der gewünschten Ankunftszeit (Fahrzeit mit stat. Verkehrsdaten)

- **time**

Datum und Uhrzeit der zu berechnenden Route; jjjj-mm-ttThh:mm

Beispiel:

```
curl -X POST
'https://sg.geodatenzentrum.de/web_ors/v2/directions_st/driving-
car/geojson' -H 'Content-Type: application/json; charset=utf-8' -H
'Accept: application/json, application/geo+json, application/gpx+xml,
img/png; charset=utf-8' -d '{"departure": "2021-06-
14T08:00", "coordinates": [[8.8368946,49.8949312], [9.4924096,51.3154546]]}'
```

Antwort siehe [3.3.2](#)

## 3.5 Erreichbarkeitsanalyse

Die Funktionen „**isochrones**“ erlauben die Berechnung von Erreichbarkeiten für einen mit Koordinaten beschriebenen Punkt mit den Parametern Zeit und Entfernung.

### 3.5.1 Anfrage

Erreichbarkeitsanalyse für Zeit/ Distanz

```
curl -X POST \
'https://sg.geodatenzentrum.de/web_ors/v2/isochrones/driving-car/geojson' \
-H 'Content-Type: application/json; charset=utf-8' \
-H 'Accept: application/json, application/geo+json, application/gpx+xml,
img/png; charset=utf-8' \
-d
'{"locations": [[8.681495,49.41461], [8.686507,49.41943]], "range": [30,60], "ra
nge_type": "time"}'
```

Parameter	Wert	Beschreibung
<b>profile</b>	driving-car cycling-regular foot-walking driving-hgv wheelchair	Geschwindigkeitsprofile Auto, Fahrrad, Fußgänger, Schwerlastverkehr oder Rollstuhl



<b>locations</b>	lon, lat	Ein mit Koordinaten (Breite, Länge) beschriebener Punkt, Die Koordinatenangaben erfolgen im Koordinatensystem WGS84 in Dezimalschreibweise getrennt durch ein Komma (.). Die Verwendung mehrerer Punkte ist möglich.
<b>range</b>	(array) [number] z.B. [300, 600]	Maximale Reichweite der Analyse in Sekunden für die Zeit und in Metern für die Entfernung oder eine durch Kommata getrennte Liste spezifischer Reichweitenwerte. Die Bereiche sind für alle Standorte gleich.
<b>attributes</b>	(array) [text] area  reachfactor  total_pop	Liste der Isochronenattribute Rückgabe der Flächengröße  Ermittlung des Erreichbarkeitsfactors  Rückgabe der Bevölkerungsstatistik
<b>id</b>	text	Beliebige Identifizierungszeichenfolge der Anfrage, die sich in den Metainformationen widerspiegelt
<b>intersections</b>	true, false	Gibt an, ob sich schneidende Polygone zurückgegeben werden sollen.
<b>interval</b>	number z.B. 100	Intervall der Isochronen oder Äquidistanten. Wird nur verwendet, wenn Erreichbarkeitsanalysen zu einem Punkt berechnet werden. Wert in Sekunden für Zeit und Meter für Entfernung
<b>location_type</b>	time / distance integer	time= Zeitangabe in Sekunden (max. 10.800) distance= Entfernungsangabe in Meter (max. 400.000)
<b>options</b>	avoid_borders all/ controlled/ none  avoid_countries (array) [number] z.B. [11, 193]  avoid_features (array) [text] Highway/ toll-ways/ ferries z.B. ["highways"]	alle für keinen Grenzübertritt. kontrolliert, um offene Grenzen zu überschreiten, aber kontrollierte Grenzen zu vermeiden. Nur für driving- Profile.  Liste der Länder, die von der Streckenführung mit Fahrprofilen auszuschließen sind. Kann zusammen mit 'avoid_borders' verwendet werden: 'controlled'. [ 11, 193 ] würde Österreich und die Schweiz ausschließen. Eine Liste von Ländern und Anwendungsbeispielen finden Sie <a href="#">hier</a> . Es können auch ISO-Standard-Ländercodes anstelle der numerischen IDs verwendet werden, z. B. DE oder DEU für Deutschland.  Liste der zu vermeidenden Verkehrstypen.

	<p>avoid_polygons geojson-Objekt</p> <p>round_trip (object) z.B. { "length":10000 , "points":5}</p>	<p>Umfasst die für die Route zu vermeidenden Gebiete. Formatiert in GeoJSON entweder als Polygon- oder Multipolygon-Objekt</p> <p>Optionen, die auf Hin- und Rückfahrtstrecken anzuwenden sind.</p>
<b>range_type</b>	<p>string time/ distance</p>	Gibt die Erreichbarkeitsart der Isochronen an
<b>smoothing</b>	<p>number z.B. 25</p>	<p>Wendet einen Generalisierungsgrad auf die erzeugten Isochronenpolygone an, der als smoothing_factor zwischen 0 und 100,0 liegt.</p> <p>Die Generalisierung wird durch die Bestimmung der maximalen Länge einer Verbindungslinie zwischen zwei Punkten auf der Außenseite eines enthaltenen Polygons erzeugt.</p> <p>Wenn der Abstand größer als ein Schwellenwert ist, wird die Linie zwischen den beiden Punkten entfernt und eine kleinere Verbindungslinie zwischen anderen Punkten verwendet.</p> <p>Beachten Sie, dass die minimale Länge dieser Verbindungslinie ~1333m beträgt. Wenn also der smoothing_factor einen geringeren Abstand als diesen ergibt, wird der minimale Wert verwendet.</p> <p>Der Schwellenwert wird bestimmt als (maximaler_Radius_der_Isochrone / 100) * smoothing_factor.</p> <p>Ein Wert, der näher an 100 liegt, führt daher zu einer allgemeineren Form.</p>
<b>area_units</b>	<p>Text m/ km/ mi</p>	<p>Gibt die Flächeneinheit an. Voreinstellung: m.</p>
<b>units</b>	<p>Text m/ km/ mi</p>	<p>Gibt die Entfernungseinheiten nur an, wenn range_type auf distance gesetzt ist. Voreinstellung: m.</p>

### 3.5.2 Antwort

Parameter	Beschreibung
<b>FeatureCollection</b>	features
<b>bbox</b>	Bounding-Box der Erreichbarkeitsanalyse

<b>features</b>	<p>group_index: Gruppierungsindex Integer</p> <p>value: Wert des betreffenden Umringspolygons (time oder distance)</p> <p>center: Koordinate des Ausgangspunktes der Erreichbarkeitsanalyse</p> <p>geometry:</p>
<b>geometry</b>	<p>Polygon-Koordinaten der Umringe der einzelnen Ranges</p> <p>coordinates": [[xx,yy],[xx,yy]...]</p>
<b>bbox</b>	<p>Bounding-Box der Route</p>
<b>metadata</b>	<p>attribution: Quellenangaben</p> <p>service. Servicetyp z.B. isochrones</p> <p>timestamp: Zeitstempel der Routinganfrage</p> <p>query: json-Object { profile: "driving-car", format: "geojson", locations: [...], ranges: [...] }</p> <p>locations: Koordinaten der Erreichbarkeitsanalysen-Ursprünge</p> <p>range: array z.B. [ 225, 450, 675, 900 ]</p> <p>range_type: time / distance</p> <p>engine Object: Angaben zur RoutingEngine, Version, Erstellungsdatum der Software, Erstellungsdatum der Graphen { version: "6.7.0", build_date: "2021-06-08T13:11:04Z", graph_date: "2021-06-16T12:52:16Z" } version "6.7.0" build_date: Erstellungsdatum der Software graph_date: Erstellungsdatum der Graphen</p>

Antwort:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "group_index": 0,
        "value": 30.0,
        "center": [8.681495737068015, 49.41461001633336]
      },
      "geometry": {
        "coordinates": [
          [
            [8.679691, 49.414567],
            [8.679742, 49.414211],
            [8.680804, 49.413608],
            [8.681095, 49.413431],
            [8.681114, 49.413426],
            [8.681378, 49.413654],
            [8.681676, 49.414613],
            [8.68167, 49.41472],
            [8.681645, 49.415192],
            [8.681621, 49.415664],
            [8.681616, 49.415754],
            [8.681257, 49.415736],
            [8.679691, 49.414567]
          ]
        ],
        "type": "Polygon"
      }
    },
    {
      "type": "Feature",
      "properties": {
        "group_index": 0,
        "value": 60.0,
        "center": [8.681495737068015, 49.41461001633336]
      },
      "geometry": {
        "coordinates": [
          [
            [8.677969, 49.414496],
            [8.677972, 49.414136],
            [8.679584, 49.412808],
            [8.681336, 49.413299],
            [8.681424, 49.413322],
            [8.68213, 49.413521],
            [8.682836, 49.413721],
            [8.683252, 49.415541],
            [8.683259, 49.415901],
            [8.68195, 49.416897],
            [8.681604, 49.416996],
            [8.679644, 49.415894],
            [8.677969, 49.414496]
          ]
        ],
        "type": "Polygon"
      }
    },
    {
      "type": "Feature",
      "properties": {
        "group_index": 1,
        "value": 30.0,
        "center": [8.686506462648579, 49.41942995674008]
      },
      "geometry": {
        "coordinates": [
          [
            [8.686112, 49.420532],
            [8.686215, 49.419994],
            [8.686318, 49.419455],
            [8.686333, 49.419379],
            [8.687047, 49.418693],
            [8.687402, 49.418754],
            [8.688247, 49.419431],
            [8.688204, 49.419788],
            [8.686465, 49.4206],
            [8.686112, 49.420532]
          ]
        ],
        "type": "Polygon"
      }
    },
    {
      "type": "Feature",
      "properties": {
        "group_index": 1,
        "value": 60.0,
        "center": [8.686506462648579, 49.41942995674008]
      },
      "geometry": {
        "coordinates": [
          [
            [8.685157, 49.420986],
            [8.685216, 49.420631],
            [8.687276, 49.417477],
            [8.687633, 49.417524],
            [8.689935, 49.419627],
            [8.689894, 49.419985],
            [8.687209, 49.421269],
            [8.685987, 49.421696],
            [8.685679, 49.421509],
            [8.685157, 49.420986]
          ]
        ],
        "type": "Polygon"
      }
    }
  ],
  "bbox": [8.677969, 49.412808, 8.689935, 49.421696],
  "metadata": {
    "attribution": "openrouteservice.org | OpenStreetMap contributors",
    "service": "isochrones",
    "timestamp": 1586248649888,
    "query": {
      "locations": [
        [8.681495, 49.41461],
        [8.686507, 49.41943]
      ],
      "range": [30.0, 60.0],
      "range_type": "time",
      "engine": {
        "version": "6.0.0",
        "build_date": "2020-03-02T08:58:53Z",
        "graph_date": "2020-03-03T11:29:47Z"
      }
    }
  }
}
```

### 3.5.3 Erreichbarkeitsanalysen-Restriktionen

maximale Anzahl Standorte= 5

maximale Anzahl Intervalle= 10

größtmöglicher Isochronenbereich= 10800 Sekunden (180 Minuten)

Distanzbereich maximal 400.000 Meter (400 Km)

## 3.6 Erreichbarkeitsanalyse mit statistischen Verkehrsflussinformationen

Für die Profile driving-car (Pkw) und driving-hvg (Schwerlastverkehr) besteht die Option Isochronen mit statistischen Verkehrsflussinformationen zu berechnen. Hierbei fließen die statistisch verifizierten, zu erwartenden Verkehrsflüsse der letzten 2 Jahre in die Routenberechnung ein. Die statistischen Verkehrsflussinformationen liegen für Bundesautobahnen, Bundesstraßen,

Landes- u. Kreisstraßen, sowie zum Teil für Gemeindestraßen vor. Die aktuell hinterlegten Verkehrsflussinformationen basieren auf Verkehrsdaten der Jahre 2018-2019.

Die Funktion Routingdienstes mit statistischen Verkehrsflussinformationen wird dem mit Zusatz **\_st** in der URL angesprochen. (Dieser Dienst steht ausschließlich Bundesbehörden zur Verfügung)

### Url des Routingsservers mit statistischen Verkehrsflussinformationen

```
server-address = sg.geodatenzentrum.de/web_ors/v2/isochrones_st/[profil  
car/hgv]
```

Zur Berechnung von Isochronen mit statistischen Verkehrsflussinformationen werden im Request folgende zusätzliche die Parameter angegeben:

- **location\_type**
  - start**  
Berechnung von Isochronen vom Startpunkt ausgehend
  - **destination**  
Berechnung von Isochronen um den Startpunkt, Rotutenberechnung jedoch reverse zum Startpunkt hin.
- **time**  
Datum und Uhrzeit der zu berechnenden Route; jjjj-mm-ttThh:mm

Beispiel:

```
curl -X POST  
'https://sg.geodatenzentrum.de/web_ors/v2/isochrones_st/driving-car' -H  
'Content-Type: application/json; charset=utf-8' -H 'Accept:  
application/json, application/geo+json, application/gpx+xml, img/png;  
charset=utf-8' -d  
'{"locations": [[6.084726, 50.774122]], "range_type": "time", "range": [900], "loc  
ation_type": "start", "time": "2020-06-01T07:30"}'
```

Antwort siehe [3.5.2](#)

## 3.7 Matrix

Die Funktion „**matix**“ ermöglicht die Berechnung der Strecken und Fahrzeiten zwischen allen Punkten, die über den Parameter „locations“ übergeben werden. Ist eine Verbindung null konnte die betreffende Route nicht berechnet werden.

### 3.7.1 Anfrage

#### Matrixberechnung für mehrere Punkte

```
curl -X POST 'https://sg.geodatenzentrum.de/web_ors/v2/matrix/driving-car'
-H 'Content-Type: application/json; charset=utf-8' -H 'Accept:
application/json, application/geo+json, application/gpx+xml, img/png;
charset=utf-8' -d
'{"locations": [[9.70093, 48.477473], [9.207916, 49.153868], [37.573242, 55.80128
1], [115.663757, 38.106467]], "metrics": ["duration"]}'
```

Parameter	Wert	Beschreibung
<b>profile</b>	driving-car cycling-regular foot-walking driving-hgv wheelchair	Geschwindigkeitsprofile Auto, Fahrrad, Fußgänger, Schwerlastverkehr oder Rollstuhl
<b>locations</b>	[[xx, yy], [xx, yy]... ]]	Ein mit Koordinaten (Breite, Länge) beschriebene Punkte. Die Koordinatenangaben erfolgen im Koordinatensystem WGS84 in Dezimalschreibweise getrennt durch ein Komma (,)
<b>destinations</b>	array z.B. [0,3]	Eine Liste von Indizes, die sich auf die Liste der Orte (beginnend mit 0) bezieht. {index_1},{index_2},{index_N} ...] oder alle (Standard). [0,3] für den ersten und vierten Ort
<b>id</b>	text	Beliebige Identifizierungszeichenfolge der Anfrage, die sich in den Metainformationen widerspiegelt
<b>metrics</b>	array [] duration / distance	distance - Gibt die Abstandsmatrix für die angegebenen Punkte in definierten Einheiten zurück. duration - Liefert eine Dauermatrix für die angegebenen Punkte in Sekunden.
<b>metricsStrings</b>	array []	Die Nutzlast/ Kosten der Anfrage
<b>resolve_locations</b>	true, false	Gibt an, ob die angegebenen Orte aufgelöst werden oder nicht. Wenn der Parameterwert auf true gesetzt ist, enthält jedes Element in destinations und sources ein name-Element, das den Namen der nächstgelegenen Straße angibt. Die Voreinstellung ist false.
<b>sources</b>	array z.B. [0,3]	Eine Liste von Indizes, die sich auf die Liste der Orte (beginnend mit 0) bezieht. {index_1},{index_2},{index_N} ...] oder alle (Standard). Beispiel [0,3] für den ersten und vierten Ort
<b>units</b>	Text m/ km/ mi	Gibt die Entfernungseinheiten nur an, wenn range_type auf distance gesetzt ist. Voreinstellung: m.

### 3.7.2 Antwort

Parameter	Beschreibung
<b>durations / distances</b>	array [] mit Zeiten oder Strecken
<b>destinations</b>	array [] Koordinaten der Zielpunkte mit snapped_distance zum Routingfähigen Netzwerk
<b>sources</b>	array [] Koordinaten der Startpunkte punkte mit snapped_distance zum Routingfähigen Netzwerk
<b>metadata</b>	<p>attribution: Quellenangaben</p> <p>service. Servicetyp z.B. isochrones</p> <p>timestamp: Zeitstempel der Routinganfrage</p> <p>query: json-Object { profile: "driving-car", format: "geojson", locations: [...]}</p> <p>locations: Koordinaten der Matrixpunkte</p> <p>engine Object: Angaben zur RoutingEngine, Version, Erstellungsdatum der Software, Erstellungsdatum der Graphen { version: "6.7.0", build_date: "2021-06-08T13:11:04Z", graph_date: "2021-06-16T12:52:16Z" } version "6.7.0" build_date: Erstellungsdatum der Software graph_date: Erstellungsdatum der Graphen</p>

**Antwort:**

```
{ "durations":
[[0.0,5778.08,90010.47,400029.72],[5784.41,0.0,88201.11,398220.34],[89148.8
1,87199.81,0.0,313222.69],[399624.91,397675.91,312417.53,0.0]], "destina-
tions": [{"location": [9.700817,48.476406], "snapped_distance":118.92}, {"loca-
tion": [9.207773,49.153882], "snapped_distance":10.54}, {"loca-
tion": [37.572926,55.80129], "snapped_distance":19.8}, {"loca-
tion": [115.665017,38.100717], "snapped_distance":648.79}], "sources": [{"loca-
tion": [9.700817,48.476406], "snapped_distance":118.92}, {"loca-
tion": [9.207773,49.153882], "snapped_distance":10.54}, {"loca-
tion": [37.572926,55.80129], "snapped_distance":19.8}, {"loca-
```

```
tion": [115.665017, 38.100717], "snapped_distance": 648.79}], "metadata": {"attribution": "openrouteservice.org | OpenStreetMap contributors", "service": "matrix", "timestamp": 1586267880961, "query": {"locations": [[9.70093, 48.477473], [9.207916, 49.153868], [37.573242, 55.801281], [115.663757, 38.106467]], "profile": "driving-car", "responseType": "json"}, "engine": {"version": "6.0.0", "build_date": "2020-03-02T08:58:53Z", "graph_date": "2020-03-03T11:29:47Z"}}
```

### 3.7.3 Matrix-Restriktionen

maximale Anzahl Standorte (Ursprung x Ziel)= 3500 pro Anfrage

Standorte (mit dynamischen Argumenten)= 25 pro Anfrage (z.B. 5 x 5)

## 3.8 Travel Salesman Routing

Die Funktion Travel Salesman Routing ermöglicht die Berechnung der günstigsten (schnellsten) Route zum Anfahren mehrer Ziele. Dabei werden die Zielpunkte zwischen dem Start- und Endpunkt in der geeignetsten Reihenfolge ausgegeben.

Für die Berechnung eines Rings müssen Star- und Endpunkt identisch sein.

### 3.8.1 Anfrage

#### Travel Salesman Routing

```
curl -X POST 'https://sg.geodatenzentrum.de/web_ors/tsr/' --header "Content-Type: application/json" --data '{"vehicles":[{"id":0,"profile":"driving-car","start":[8.6526,49.4604],"end":[8.6526,49.4604]}],"jobs":[{"id":0,"location":[8.6691,49.4532]},{"id":1,"location":[8.6911,49.4566]}],"options":{"g":true}}'
```

Parameter	Wert	Beschreibung
<b>vehicles</b>	array [ ]	Liste an Fahrzeugen
<b>profile</b>	driving-car cycling-regular foot-walking driving-hgv	Geschwindigkeitsprofile Auto, Fahrrad, Fußgänger oder Schwerlastverkehr
<b>start</b>	lon/lat	Koordinaten Startpunkt (lon/lat) WGS84
<b>end</b>	lon/lat	Koordinaten Endpunkt (lon/lat) WGS84
<b>jobs</b>	array [{"id":0,"location":[lon/lat]}, {"id":1,"location":[lan/lat]... }]	Koordinaten der anzufahrenden Zielpunkte (lon/lat) WGS84
<b>options</b>	"g": true "t": int "l": int	Optionale Parameter "g": true = detaillierte Streckengeometrie und Entfernung berechnen



		"t": int = Anzahl der verfügbaren Threads (Standard: 4) "l": int = Beendigung des Lösungsprozesses nach 'l' Sekunden
--	--	---

### 3.8.2 Antwort

Parameter	Beschreibung
<b>code</b>	code 0= kein Error 1= interner Fehler 2= Eingabe Fehler 3= Routing-Fehler
<b>summary</b>	Informationen Gesamtroute
<b>distances oder durations</b>	[mm, mm...] oder (sek,sek...]
<b>routes</b>	Informationen Einzelroute
<b>steps</b>	Koordinaten der Routenpunkte start[lon/lat]... job[lon/lat] (Ziele) ... end[lon/lat]
<b>geometry</b>	Polylinienkodierte Routengeometrie (pro step)

Antwort:

```
{
  "code": 0,
  "summary": {
    "cost": 2857,
    "unassigned": 0,
    "service": 0,
    "duration": 2857,
    "waiting_time": 0,
    "priority": 0,
    "distance": 22069,
    "violations": [],
    "computing_times": {
      "loading": 88,
      "solving": 0,
      "routing": 11
    }
  },
  "unassigned": [],
  "routes": [
    {
      "vehicle": 0,
      "cost": 2857,
      "service": 0,
      "duration": 2857,
      "waiting_time": 0,
      "priority": 0,
      "distance": 22069,
      "steps": [
        {
          "type": "start",
          "location": [8.6526, 49.4604],
          "service": 0,
          "waiting_time": 0,
          "arrival": 0,
          "duration": 0,
          "violations": []
        },
        {
          "type": "job",
          "location": [8.6911, 49.4566],
          "id": 1,
          "service": 0,
          "waiting_time": 0,
          "job": 1,
          "arrival": 1331,
          "duration": 1331,
          "violations": []
        },
        {
          "type": "job",
          "location": [8.6691, 49.4532],
          "id": 0,
          "service": 0,
          "waiting_time": 0,
          "job": 0,
          "arrival": 2711,
          "duration": 2711,
          "violations": []
        },
        {
          "type": "end",
          "location": [8.6526, 49.4604],
          "service": 0,
          "waiting_time": 0,
          "arrival": 2857,
          "duration": 2857,
          "violations": []
        }
      ],
      "distance": 22069,
      "violations": [],
      "geometry": "sh{lhEcys@iAd-AwA~@s@\\q@VwA^uH|AiK|ByBf@aE|@q@PmAZyB`cARoDv@u@PMDYFShWRORw@}Ak@aABI@KA-GAEEKECEAMDCDOqAbcBcAkAg@i@u@aAYe@i@aAc@{@MU[ {@c@kAKa@Mm@?CaAuEESu@eEOyAG-cAC_B@o@?cA?SRgA@Y@MYLAMUq@uC^SIODGy@G_A?m@DcA?KH}A?}@CcAQwBIu-AAOEK@IwAA[Cw@EeBAGGc@OoAOc@k@kAs@e@Q_@IUMY?Y?}ABg@Hm@J[FGHKF]DY-BUBi@G}@GcCa@qCm@a@aQMGEe}EgA@}ADuB-DoC@oAc{AMeDAwADgAJ_BP}CXoBhBkHFu@DmAd@eB?wAC}B?E?e@B{@TkCBg@?i@G{AM{B@}@JgB Bc@P?RAT@j@@@XBNFLMd@?BK?E@E?S@UDYAbq@^sDJeD\\_Eh@_DlAeG^iCZwAF_@?UASyO@EW?}Dm@Fm@Tm@n@eAJa@Co@q@_DAUFUHG-JAx@NnAdz@LhAVd@Bv@GbAg@d@c@b@q@\\gAH{@@c@E_@OQOGaACu@IyBaAMYC}BkBAu@Kw@a@mAEW@[J_@NYL-IbCY~@?n@Fr@NpAHd@Gp@StAi@FKb@mBTi@NUT}RS^q@Te@Rg@VkAZeBZ_CD{@?o@UyA_@cBEg@Bg@Hc@H}NSJ^Tn@^n@`z@h@lAd@rAph@JdAFbABf@Bd@FRd@nAT-bBBZDSHQHUZq@Vc@POb@MhDs@`BOD@Bp@Rz@d@xA-bAvBh~EvE`BbAn@l@xArBN\\X^p@`@`AJ`@Vj@Hx@Rr@ZZT^l@Zt@Z^\\R^HZ?NVPd@Nl@JvAL
```

```
h@t@bA`Aj@^XlAhAR^\\fBPj@V\\VnH@BbBIXIr@A`@L~@p@b@XbAfa-  
dAj@f@`@TXZp@Tz@r@nBclAzAdNFdDoAvPQ`@GDKA-  
KIY_@uBiFsFuSsA{BkCmCkAs@iAgB[]@Z|@hAfBjAr@jClCrAz-  
BrFtStBhFX^JHJ@FEPa@nAwPGeD{AeNBmAs@oBU{@[q@UYg@a@eAk@cAgAc@Y_Aq@a@Ms@@YHcB  
Hi@CWOW]Qk@]gBS_@mAiA_@YaAk@u@cAmi@KwAom@Qe@OW[?_@I]S[_@[u@_@m@[Us@[y@Sk@Ia  
@WaAkq@a@Y_@O]yAsBo@m@aBcA_FwEwBiByAcA{@e@q@Se@CaB-  
NiDr@c@LQNwb@[p@ITIPERC[UcBe@oAG-  
SCe@Cg@GcAKeAQi@e@sAi@mAa@{@_@o@Uo@K_@ORI\\Ib@Cf@Df@^bBTxA?n@Ez@[~B[dBwjASf  
@Ud@_@p@SRU\\@OTUh@c@lBGJuAh@q@Re@FqAIs@Oo@G_A?cXMHoxK^AZDV`@lAJv@t@t@CjBB\\  
LXxB`At@H`AB-  
NFNPD^Ab@Iz@]fAc@p@e@b@cAf@w@Fe@CiAW{@MoAEy@OK@IFGT@Tp@~CBn@K`@o@dAUl@Gl@El  
@?\\DVXn@@R?TG@[vA_@hCmAdGi@~C]~DKdD_@rDCp@Ex-  
AAT?RAD?DCJe@?MLOGYck@AUAS@Q?Cb@KfBA|@LzBFzA?h@Cf@UjCCz@?d@?DB|B?vAEdBELAGt  
@iBjHYnBQ|CK~AEfA@vAlDDBzAAAnAEnCEtBA|ADfAD\\Jh@Jd@DrAA^?NG|BARE|@KlA-  
EZCREVC\\@d@Fb@r@nCNZLXHTP^Rd@j@jANb@NnAFb@@FDdBBv@@ZHvADj@@NHtAPvBBbA?|@I|  
A?JEbA?l@F~@Fx@NERHtC_@Tp@@LXmbBm@j@WXM-  
tAg@VIRCaAFANGfC}@x@YtBq@xBs@bA]XKfDiA^mfBm@lFcBz@W`@MdBc@fD}@lKiCpS}EpJ{Br  
Cs@dD_AnAYX@PDAJ@JDRFFDBH?DALKBI@MC[EKNOPIjB_@bA]nDcAJCCW?]Bi@Cy-  
ACg@t@YMkBGkAFjALjBu@XBf@BxACh@?\\BVKBOD-  
bAcA\\kB^QHONGGQAOPCLAJ@JDRFFAVAj@a@zBa@tBAR?LKTIPeZa@pBw@tD{@jDcD~M{@pDw@p  
CmAbDwAxCwCtFo@fAm@z@o@v@ED"} ] }
```

### 3.8.3 Travel Salesman Restriktionen

maximale Anzahl der Zielpunkte beträgt 50

## 4 Datenschutz

Der Routingdienst stellt Routinginformationen ohne jeden Personenbezug bereit und kann nur entsprechende Anfragen ohne jeglichen Personenbezug verarbeiten.

Der Routingdienst und ebenso die vorgelagerte IT-Infrastruktur des BKG speichern grundsätzlich keine an ihn gerichteten Anfragen mit ihren semantischen Parametern, auch nicht temporär. Die Suchparameter und die Ergebnisse der Suche werden nur flüchtig („on the fly“) verarbeitet, um die Antwort an den Client senden zu können.

Es findet ausschließlich eine synchrone Verarbeitung der Routeninformation ohne jede Zwischenspeicherung statt.

## 5 Quellenverzeichnis

1	<a href="https://openrouteservice.org">https://openrouteservice.org</a> (Letzter Aufruf am 7.2.2024)
3	Bundesamt für Kartographie und Geodäsie, Richard-Strauss-Allee 11, 60598 Frankfurt/Main
9	<i>GeoJSON</i> . <a href="http://geojson.org/">http://geojson.org/</a>

## 6 Kontakt

### 6.1 Freischaltung

Für die Freischaltung des Dienstes richten Sie sich bitte an:

Bundesamt für Kartographie und Geodäsie  
Dienstleistungszentrum  
Karl-Rothe-Straße 10-14  
04105 Leipzig  
E-Mail: [dlz@bkg.bund.de](mailto:dlz@bkg.bund.de)  
Tel.: 0341/5634-333  
Fax: 0341/5634-415

### 6.2 Technische Anfragen

Technische Anfragen richten Sie bitte an:

Dienstleistungszentrum  
E-Mail: [dlz@bkg.bund.de](mailto:dlz@bkg.bund.de)  
Tel.: 0341/5634-333  
Fax: 0341/5634-415