



PUBLIC

SAP HANA Spatial Services 1.0

2025-10-30

SAP HANA Spatial Services Reference

Content

1	Introduction.	3
2	Prerequisites.	4
2.1	SAP BTP Enterprise Account Based on the Consumption-Based Commercial Model.	4
2.2	Using SAP HANA Spatial Services in the Cloud Foundry Environment.	5
2.3	Using SAP HANA Spatial Services in the Kyma Environment.	5
2.4	GeoJSON Format.	6
3	Endpoints.	7
4	SAP HANA Spatial Services.	9
4.1	Mapping API.	9
4.2	Geocoding API.	24
4.3	Routing APIs.	30
	Single Routing API.	31
	Matrix Routing API.	42
	Using the various Parameter.	60
4.4	API Transaction Costs.	66
4.5	API Responses.	68
5	Frequently Asked Questions for SAP HANA Spatial Services.	69
6	Accessibility Features in SAP HANA Spatial Services.	77

1 Introduction

SAP HANA spatial services provide a set of services running on SAP Business Technology Platform (SAP BTP). These services focus on providing simple access through an unified interface to geocoding, mapping and routing functionalities integrated from third-party service providers. Thus, customers benefit from easy-to-consume APIs, and finally an efficient way to infuse business data.

SAP HANA spatial services helps you to quickly develop location-aware business solutions that can be readily utilized in SAP®, custom or third party applications. By abstracting the back-end service provider, applications can leverage a single integration to access location information with flexibility to adapt to local needs.

📌 Note

When using SAP HANA spatial services APIs relying on the established contract agreements between SAP and **Here**, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Please be aware that this does not generally apply when bringing your own license (BYOL model). In this case, the terms of your own license apply.

Please check the details of the service agreement available at [PROFESSIONAL SERVICES AGREEMENT](#) 

Environment

SAP HANA spatial services is available the following environments:

- Cloud Foundry Enviroment
- Kyma Environment

2 Prerequisites

Before using the set of services provided by SAP HANA spatial services, you must fulfill the following prerequisites.

[SAP BTP Enterprise Account Based on the Consumption-Based Commercial Model \[page 4\]](#)

SAP HANA spatial services can be used with enterprise accounts based on the consumption-based model in two type of plans.

[Using SAP HANA Spatial Services in the Cloud Foundry Environment \[page 5\]](#)

If you want to consume SAP HANA spatial services programmatically within the Cloud Foundry environment, you not only need a global account for Cloud Foundry, you also need to configure a subaccount and space from which to access the services.

[Using SAP HANA Spatial Services in the Kyma Environment \[page 5\]](#)

You can use SAP HANA spatial services programmatically within the Kyma environment.

[GeoJSON Format \[page 6\]](#)

Many SAP HANA spatial services APIs respond with or expect objects in GeoJSON format. GeoJSON is a format for encoding a variety of geographic data structures.

2.1 SAP BTP Enterprise Account Based on the Consumption-Based Commercial Model

SAP HANA spatial services can be used with enterprise accounts based on the consumption-based model in two type of plans.

- Cloud Platform Enterprise Agreement (CPEA)
- Pay-As-You-Go for SAP BTP

For more information, see [Basic Platform Concepts](#) and [What Is the Consumption-Based Commercial Model?](#).

For more information about onboarding SAP HANA spatial services, see [SAP HANA Spatial Services Onboarding Guide](#).

Note

In order to access the here documented APIs, you will need to use the service plan **lite**.

2.2 Using SAP HANA Spatial Services in the Cloud Foundry Environment

If you want to consume SAP HANA spatial services programmatically within the Cloud Foundry environment, you not only need a global account for Cloud Foundry, you also need to configure a subaccount and space from which to access the services.

You must create a service instance for each spatial service of interest from the respective subaccount. You can also generate service keys for these service instances, to enable local clients, apps in other spaces, or entities outside your deployment to access the services.

For more details about how to use services in Cloud Foundry, see the *Using Services in the Cloud Foundry Environment* section of the official [SAP Business Technology Platform \(SAP BTP\) documentation](#).

For direct programmatical API consumption, a valid authorization token must be passed as part of the API request. Cloud Foundry uses the OAuth2 standard for authorization and authentication of user requests.

SAP HANA spatial services support X.509 client certificates. How to implement X.509 client certificates, see the optional steps during the setup of the instance and the service key in [Setting up a New Instance](#).

A valid authorization token can be requested from the authorization entity listed in the service key under `url` using the individual `clientId` and `clientSecret`.

Example using curl to request OAuth access token:

```
curl '<<authority_url>>/oauth/token' -i -u '<<clientId>>:<<clientSecret>>'
-X POST -H 'Content-Type: application/x-www-form-urlencoded' -H 'Accept:
application/json' -d 'grant_type=client_credentials&response_type=token'
```

Tokens received from the authorization entity are only valid for a limited time and need to be requested again after expiration. Once a valid access token has been received, the service itself can finally be called programmatically. The obtained `access_token` needs to be specified within the request header:

```
Authorization: Bearer <<access_token>>
```

2.3 Using SAP HANA Spatial Services in the Kyma Environment

You can use SAP HANA spatial services programmatically within the Kyma environment.

For more details about how to use services in Kyma, see the *Using SAP BTP Services in the Kyma Environment* section of the official [SAP Business Technology Platform \(SAP BTP\) documentation](#).

2.4 GeoJSON Format

Many SAP HANA spatial services APIs respond with or expect objects in GeoJSON format. GeoJSON is a format for encoding a variety of geographic data structures.

You can find the GeoJSON specification here: <https://tools.ietf.org/html/rfc7946>  .

3 Endpoints

📘 Note

Please find a detailed technical API description including Swagger documentation and try-out functionality at <https://api.sap.com/package/SAPHANASpatialServices/rest> 📄

Mapping API

The Mapping API provides the following endpoints:

- Raster Mapping Services
 - `GET /mapping/v1/{provider}/tiles/{resource}/{mapId}/{scheme}/{zoom}/{column}/{row}/{size}/{format}?lang={lang}&api_key={api_key}&politicalView={politicalView}`
- Vector Mapping Services
 - Consumer Endpoint:
 - `GET /mapping/v1/esri/styles/vector/{scheme}?api_key={api_key}`
 - Technical Endpoints:
 - `GET /mapping/v1/esri/tiles/vector/{basemap}?api_key={api_key}`
 - `GET /mapping/v1/esri/tiles/vector/{basemap}/{zoom}/{column}/{row}?api_key={api_key}`
 - `GET /mapping/v1/esri/glyphs/vector/{basemap}/font/{fontstack}/{range}?api_key={api_key}`
 - `GET /mapping/v1/esri/sprites/vector/{id}/sprite{size}.{format}?api_key={api_key}`
- Mapping Service Health
 - `GET /mapping/v1/health`

Single Routing API

The Single Routing API provides the following endpoints:

- `POST /routing/v1/route`
- `GET /routing/v1/health`

Matrix Routing API

The Matrix Routing API provides the following endpoints:

- [POST /routing/v1/matrix](#)
- [POST /routing/v2/matrix](#)
- [GET /routing/v1/health](#)
- [GET /routing/v2/health](#)

Geocoding API

The Geocoding API provides the following endpoints:

- [POST /geocoding/v1/geocode](#)
- [GET /geocoding/v1/health](#)

4 SAP HANA Spatial Services

The following topics provide detailed information about the services provided by SAP HANA spatial services.

[Mapping API \[page 9\]](#)

The web mapping service provides REST services to image tiles required for building a base map. You can use these map tiles in Geographic Information System (GIS) and other spatial applications as a background map for the visualization of georeferenced data.

[Geocoding API \[page 24\]](#)

The Geocoding API provides REST services to transform multiple addresses at once into geographical positions using different providers via a unified interface.

[Routing APIs \[page 30\]](#)

SAP HANA Spatial Services supports both single and matrix routing APIs from different providers via a unified interface.

[API Transaction Costs \[page 66\]](#)

The individual SAP HANA spatial services APIs differ in their complexity involved in performing the respective calculations. Therefore, single requests are metered and billed differently.

[API Responses \[page 68\]](#)

The following describes the response codes that can result from any call of the spatial services APIs.

4.1 Mapping API

The web mapping service provides REST services to image tiles required for building a base map. You can use these map tiles in Geographic Information System (GIS) and other spatial applications as a background map for the visualization of georeferenced data.

Supported Providers

In the following table, you can find the official API documentation of the respective provider.

Provider	Documentation Link
Here	https://www.here.com/docs/category/raster-tile-api-v3 ➡
Esri	Esri Vector Mapping API: https://developers.arcgis.com/rest/services-reference/enterprise/vector-tile/ ➡ Esri Vector Styles: https://developers.arcgis.com/rest/basemap-styles/ ➡
PTV	https://developer.myptv.com/documentation/raster-maps-api ➡
TomTom	https://developer.tomtom.com/map-display-api/documentation/raster/raster-service ➡
Trimble	https://developer.trimblemaps.com/restful-apis/mapping/map-images/map-tile/ ➡

Note

Please find a detailed technical API description including Swagger documentation and try-out functionality at https://api.sap.com/api/mapping_api/overview

Note

When using the Mapping API relying on the established contract agreements between SAP and **Here**, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Please be aware that this does not generally apply when bringing your own license (BYOL model). In this case, the terms of your own license apply.

Please check the details of the service agreement available at [PROFESSIONAL SERVICES AGREEMENT](#)

The Mapping API provides the following endpoints:

- Raster Mapping Services
 - `GET /mapping/v1/{provider}/tiles/{resource}/{mapId}/{scheme}/{zoom}/{column}/{row}/{size}/{format}?lang={lang}&api_key={api_key}&politicalView={politicalView}`
- Vector Mapping Services
 - Consumer Endpoint:
 - `GET /mapping/v1/esri/styles/vector/{scheme}?api_key={api_key}`
 - Technical Endpoints:
 - `GET /mapping/v1/esri/tiles/vector/{basemap}?api_key={api_key}`
 - `GET /mapping/v1/esri/tiles/vector/{basemap}/{zoom}/{column}/{row}?api_key={api_key}`
 - `GET /mapping/v1/esri/glyphs/vector/{basemap}/font/{fontstack}/{range}?api_key={api_key}`
 - `GET /mapping/v1/esri/sprites/vector/{id}/sprite{size}.{format}?api_key={api_key}`
- Mapping Service Health
 - `GET /mapping/v1/health`

Raster Mapping Service Definition

The raster mapping service can be used to visualize map tiles as a base map layer in a GIS system or a custom web application.

For provider **Here**, all available path parameters from the **Here Map Tile** API are supported, as well as custom display options, multiple tile resources and color schemes.

For provider **PTV**, all available path parameters are supported.

To ensure API consistency, all available path parameters are mapped to a generalized naming scheme which is valid for all providers.

For provider **TomTom**, all available path parameters are supported.

For provider **Trimble**, all available path parameters are supported.

Note

When relying on the established contract agreements between SAP and **Here**, using the provider **Here** to retrieve raster map tiles for active routing and track and trace scenarios is not allowed.

For providers **PTV**, **TomTom**, and **Trimble**, currently only the *Bring-Your-Own-License* model is supported. Therefore, the terms of your own license apply.

Endpoint:

```
GET /mapping/v1/{provider}/tiles/{resource}/{mapId}/{scheme}/{zoom}/{column}/{row}/{size}/{format}?lang={lang}&api_key={api_key}&politicalView={politicalView}
```

Parameters:

- **provider**: required (string)
Defines the provider to retrieve mapping data from.
Supported values: `here`, `ptv`, `tomtom`, `trimble`.
- **resource**: required (string)
Defines the tile type, which specifies which kind of data should be displayed on the map. The following table provides information about which provider supports which resource values.

Provider	Supported Values	Comments
Here	maptile, labeltile	Find a complete list of additional provider-specific resources in the Here documentation .
PTV	maptile, streettile, labeltile	
TomTom	maptile, streettile, labeltile	Note that to request a satellite tile, the resource <code>maptile</code> in conjunction with the <code>satellite.day</code> or <code>satellite.night</code> schemes needs to be used.
Trimble	maptile	

- **mapId**: required (string)
Specifies the map version. Allows to retrieve a specific version of mapping data. Usually, the newest data is of interest. The only supported value is `newest`.
- **scheme**: required (string)
Defines the tile scheme, which represents the style of a tile resource. The following table provides information about which provider supports which scheme values.

Provider	Supported Values	Comments
Here	normal.day, normal.night, satellite.day	Find a complete list of additional provider-specific view schemes in the Here documentation .

Provider	Supported Values	Comments
PTV	normal.day, normal.night, satellite.day	
TomTom	normal.day, normal.night, satellite.day	
Trimble	normal.day, normal.night, satellite.day, satellite.night	

- **zoom:** required (integer)
Defines the zoom level of the map image.
- **column:** required (integer)
Defines the tile column number.
- **row:** required (integer)
Defines the tile row number.
- **size:** required (integer)
Defines the size of the returned map tile in pixels per side. The following table provides information about which provider supports which size values.

Provider	Supported Values	Comments
Here	256, 512	
PTV	256 to 1024	Sizes beyond 256 can be any number up to 1024 and are only supported for non-satellite.day schemes.
TomTom	256, 512	512 is only supported for the normal.day scheme in conjunction with maptile or labeltile resource
Trimble	256, 512	512 is only supported for the normal.day and normal.night schemes.

- **format:** required (string)
Defines the output format of the map tiles. The following table provides information about which provider supports which format values. Also note that different formats use different data compression algorithms resulting in smaller image sizes. Choosing a smaller image format can result in improved performance.

Provider	Supported Values
Here	png, png8, jpg
PTV	png
TomTom	png, jpg
Trimble	png, jpg

Query Parameters:

- **lang**: optional (string)
Defines the language of map tiles. If the parameter is not declared or the value is not supported then the default language of the provider will be used. It uses MARC three-letter language codes such as `fre` for French and `ger` for German.

```
lang=fre
```

- **api_key**: optional (string)
Defines the API key to be used to authenticate at the provider. Customers with license agreements with the provider specify their own API key by setting this parameter. If the parameter is omitted or set to `SAP-KEY`, SAP will use established contract agreements with the respective provider for the request. The following table provides information about which provider supports which `api_key` values.

Provider	Supported Values	Comments
Here	SAP-KEY, <customer's own license key>	Only for provider Here, the OEM license model is currently supported.
PTV	<customer's own license key>	Only the <i>Bring-your-own-license</i> model is supported. The <code>api_key</code> parameter is required.
TomTom	<customer's own license key>	Only the <i>Bring-your-own-license</i> model is supported. The <code>api_key</code> parameter is required.
Trimble	<customer's own license key>	Only the <i>Bring-your-own-license</i> model is supported. The <code>api_key</code> parameter is required.

```
api_key=U53R-ap1-key-AaBb-CcDdEeF_fGgHhIiJjKk
```

- **politicalView**: optional (string)
Defines the political view of map tiles representing certain country's view. List of supported countries is maintained by provider. The following table provides information about which provider supports which `politicalView` values.

Provider	Supported Values	Comments
Here	ARE, ARG, BHR, EGY, IND, KWT, MAR, OMN, PAK, QAT, RUS, SAU, SRB, TUR, VNM	
PTV		<code>politicalView</code> is not supported by PTV
TomTom	Unified, Arabic, ARG, CHN, IND, ISR, KOR, MAR, PAK, RUS, TUR	

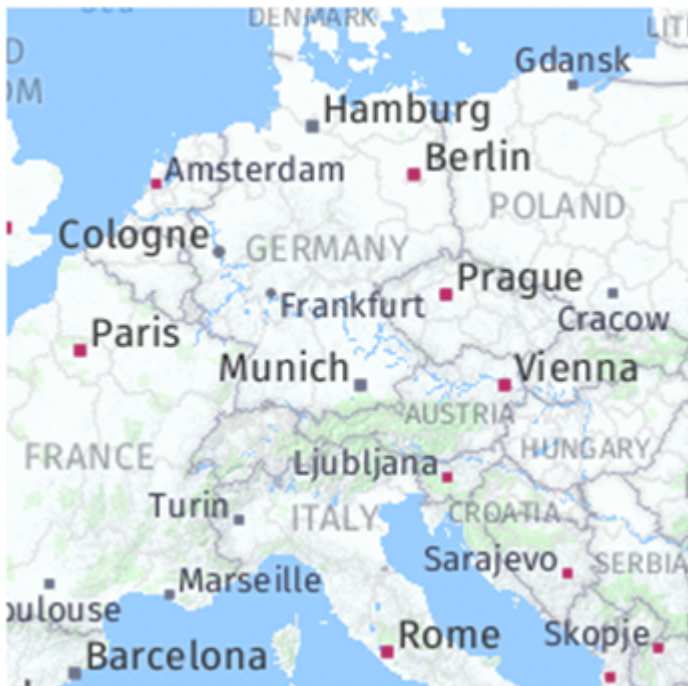
Provider	Supported Values	Comments
Trimble		politicalView is not supported by Trimble

```
politicalView=ARG
```

Example 1 - Request daytime map tile of provider Here

```
/mapping/v1/here/tiles/maptile/newest/normal.day/4/8/5/256/png
```

Example 1- Response



Example 2 - Request daytime map tile of provider TomTom

```
/mapping/v1/tomtom/tiles/maptile/newest/normal.day/4/8/5/256/png
```

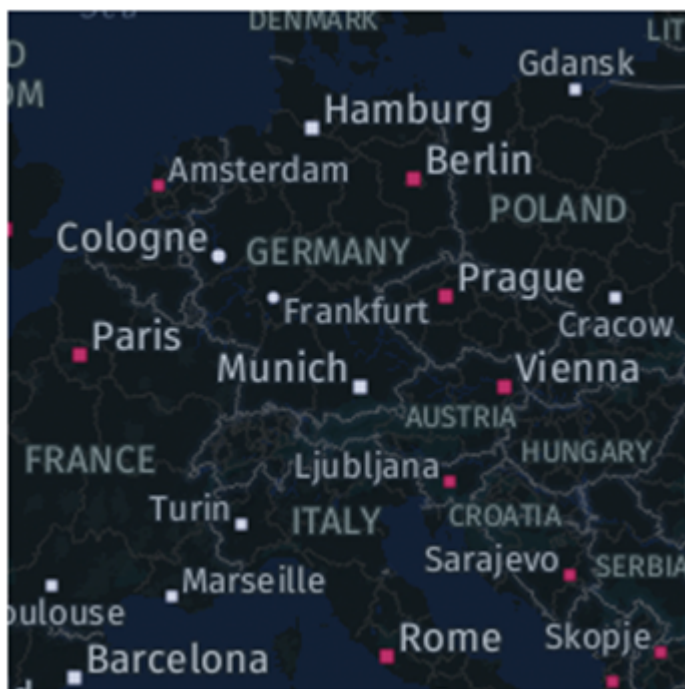
Example 2 - Response



Example 3 - Request night-time map tile

```
/mapping/v1/here/tiles/maptile/newest/normal.night/4/8/5/256/png
```

Example 3 - Response



Example 4 - Request hybrid map tile

```
/mapping/v1/here/tiles/maptile/newest/hybrid.day/4/8/5/256/png
```

Example 4 - Response



Example 5 - Request satellite map tile

```
/mapping/v1/here/tiles/maptile/newest/satellite.day/4/8/5/256/png
```

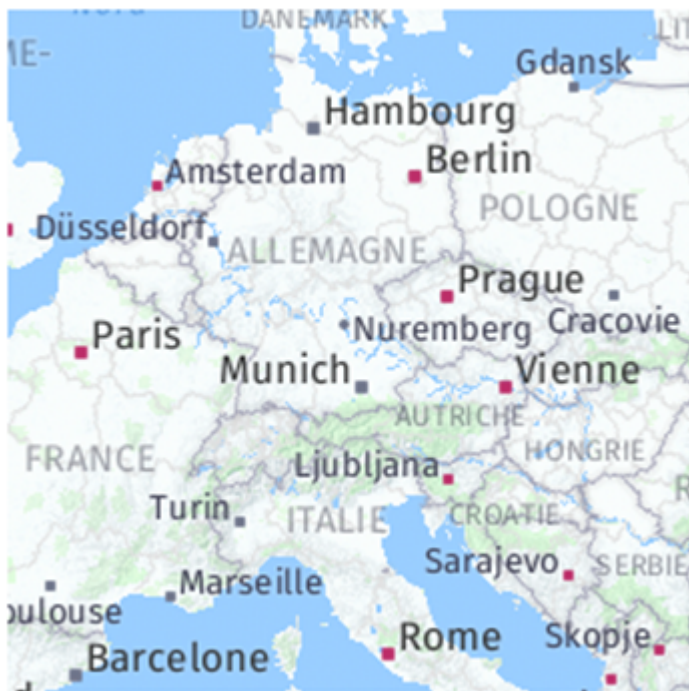
Example 5 - Response



Example 6 - Request map tile with French labels and user specific API key

```
/mapping/v1/here/tiles/maptile/newest/normal.day/4/8/5/256/png?  
lang=fr&api_key=U53R-ap1-key-AaBb-CcDdEeF_fGgHhIiJjKk
```

Example 6 - Response



Example 7: Request map tile with Argentinian political view on the ice field between Chile and Argentina

```
/mapping/v1/here/tiles/maptile/newest/normal.day/5/9/21/256/png?politicalView=ARG
```

Example 7 – Response



Vector Mapping Service Definition

The vector mapping service can be used to visualize vector map tiles with predefined styles as a base map layer in a GIS system or a custom Web application. The service exposes endpoints to fetch styles, single vector tiles, glyphs, and sprites.

The provided endpoints are categorized as a Consumer Endpoint which typically is used first to request a desired style, and as Technical Endpoints which are typically used by a map rendering engine to finally depict the desired vector maps.

Note

Currently, only **Esri** is supported as provider for vector map data.

Also, you must bring your own license with **Esri** to use this service.

Consumer Endpoint:

```
GET /mapping/v1/esri/styles/vector/{scheme}?api_key={api_key}
```

Endpoint to request vector style definitions. Provides a style definition to enable clients to render vector tiles in the desired style.

The current implementation only supports a subset of basemap styles from the original [Esri Basemap-Layer-Service](#).

Parameters:

- **scheme**: required (string)

Defines the view scheme. Available basemap styles are:

- `arcgis.dark.gray`
- `arcgis.dark.gray.base`
- `arcgis.dark.gray.labels`
- `arcgis.light.gray`
- `arcgis.light.gray.base`
- `arcgis.light.gray.labels`
- `arcgis.navigation`
- `arcgis.navigation.night`
- `arcgis.streets`
- `arcgis.streets.night`
- `osm.dark.gray`
- `osm.dark.gray.base`
- `osm.dark.gray.labels`
- `osm.light.gray`
- `osm.light.gray.base`
- `osm.light.gray.labels`
- `osm.streets`

Query Parameters:

- **api_key**: required (string)
API key that is used for authentication with the selected provider.

Example - Request vector style definitions

```
/mapping/v1/esri/styles/vector/arcgis.streets?api_key={api_key}
```

Output values:

- **version**: number
Represents version of the style specification ([Mapbox GL Style specification](#) 📖)
- **sprite**: string
Path to Vector Tile Sprite resources.
- **glyphs**: string
Path to Vector Tile Font resources.
- **sources**:
An object that contains details about the style source and corresponding vector tile.
- **layers**:
List containing style information about layers such as various types of roads, buildings, etc., for the corresponding vector tile.

Example - Response vector style definitions

```
{
  "version": 8,
  "sprite": "https://spatialservices.cfapps.eu10.hana.ondemand.com/mapping/v1/esri/sprites/vector/de26a3cf4cc9451298ea173c4b324736/sprite?api_key=<api_key>",
  "glyphs": "https://spatialservices.cfapps.eu10.hana.ondemand.com/mapping/v1/esri/glyphs/vector/arcgis/font/{fontstack}/{range}.pbf?api_key=<api_key>",
  "sources": {
```

```

      "esri": {
        "type": "vector",
        "url": "https://spatialservices.cfapps.eu10.hana.ondemand.com/mapping/v1/esri/tiles/vector/arcgis?api_key=<api_key>",
        "minzoom": 0,
        "maxzoom": 16,
        "tiles": [
          "https://spatialservices.cfapps.eu10.hana.ondemand.com/mapping/v1/esri/tiles/vector/arcgis/{z}/{x}/{y}?api_key=<api_key>"
        ],
        "copyrightText": "Sources: Esri, Here, Garmin, FAO, NOAA, USGS, © OpenStreetMap contributors, and the GIS User Community",
        "attribution": "Sources: Esri, Here, Garmin, FAO, NOAA, USGS, © OpenStreetMap contributors, and the GIS User Community"
      }
    },
    "layers": [
      {
        "id": "Land/Not ice",
        "type": "fill",
        "source": "esri",
        "source-layer": "Land",
        "filter": [
          "=",
          "_symbol",
          0
        ],
        "minzoom": 0,
        "layout": {},
        "paint": {
          "fill-color": "#f7f6d5"
        }
      },
      ...
    ]
  }
}

```

Technical Endpoint:

```
GET /mapping/v1/esri/tiles/vector/{basemap}?api_key={api_key}
```

Endpoint to request basic information about the selected set of vector tiles (basemap).

This URL is provided as part of the sources property of a vector style definition and will usually be handled by a client-side map rendering engine.

Parameters:

- **basemap**: required (string)
Defines the basemap which provides the vector tiles. The value is either arcgis or osm.

Query Parameters:

- **api_key**: required (string)
API key that is used for authentication with the selected provider.

Technical Endpoint:

```
GET /mapping/v1/esri/tiles/vector/{basemap}/{zoom}/{column}/{row}?api_key={api_key}
```

Endpoint to request **Esri** vector tiles.

This URL is provided as part of the sources property of a vector style definition and will usually be handled by a client-side map rendering engine.

Parameters:

- **basemap**: required (string)
Defines the basemap of the request vector tile. Possible value is either arcgis or osm.
- **zoom**: required (integer)
Zoom level of the vector tile. Example: 4
- **column**: required (integer)
Tile column number. Example: 8
- **row**: required (integer)
Tile row number. Example: 5

Query Parameters:

- **api_key**: required (string)
API key that is used for authentication with the selected provider.

Technical Endpoint:

```
GET /mapping/v1/esri/glyphs/vector/{basemap}/font/{fontstack}/{range}.pbf?  
api_key={api_key}
```

Endpoint to request glyphs (fonts) corresponding to a particular basemap. Glyphs are used for rendering texts on the vector map.

This URL is provided as part of the glyphs property of a vector style definition and will usually be handled by a client-side map rendering engine.

Parameters:

- **basemap**: required (string)
Defines the view scheme. The value is either arcgis or osm.
- **fontstack**: required (string)
Defines the font stack. Example: Noto Sans Regular
- **range**: required (string)
Define the range of the fontstack. Example: 0-255

Query Parameters:

- **api_key**: required (string)
API key that is used for authentication with the selected provider.

Technical Endpoint:

```
GET /mapping/v1/esri/sprites/vector/{id}/sprite{size}.{format}?api_key={api_key}
```

Endpoint to request sprites, which are used to draw signs, fill polygons, or draw a point during map rendering.

This URL is provided as part of the sprites property of a vector style definition and will usually be handled by a client-side map rendering engine.

Parameters:

- **id**: required (string)
Defines the id of the requested sprites. This parameter will be provided by the vector style definition.
- **size**: optional (string)
Defines the size of the requested sprites. Example: @2x

Because the parameter **size** is optional, this endpoint can also be called without it:

```
GET /mapping/v1/esri/sprites/vector/{id}/sprite.{format}?api_key={api_key}
```

- **format:** required (string)
Defines the format of the requested sprites. Supported formats: png, json.

Query Parameters:

- **api_key:** required (string)
API key that is used for authentication with the selected provider.

Vector map examples

Below are two examples of the same map excerpt. Both are rendered on the client using the same basemap vector data using different styles received from the consumer endpoint.



Style: `arcgis.streets`



Style: `arcgis.dark.gray`

Mapping Service Health

Endpoint:

```
GET /mapping/v1/health
```

The health endpoint can be used to determine the service's current status.

The status code and the response body provide detailed information about the service's health status.

Output values:

- **providers:** optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, tomtom, trimble

Example - Response

```
{
  "status": "UP",
  "timestamp": 1690457826,
  "message": "All services are healthy",
  "details": {
    "HERE": [
      {
        "service": "maps",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "TOMTOM": [
      {
        "service": "maps",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "TRIMBLE": [
      {
        "service": "maps",
        "status": "OK",
        "message": "Service is healthy"
      }
    ]
  }
}
```

Output values:

- **status:** string
The service's overall health status. The value is "UP", if the service is healthy, and "CRITICAL" otherwise.
- **timestamp:** integer
The timestamp (Unix time) when the service reported the healthy status.
Example: 1606308415
- **message:** string
Message to explain what the reported status means.
Example: All services are healthy
- **details**
An object that contains further details about the checks that lead to the service's reported overall status.
 - **Here**
Contains the results of health checks for the respective **Here** service.
 - **service:** string
The **Here** service, in this case maps, whose status was checked (see [Here Platform Status](#)).
 - **status:** string
The status of the checked **Here** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.
 - **message**
Message to explain what the reported status means.
 - **TomTom**

Contains the results of health checks for the respective **TomTom** service.

- **service:** string
The **TomTom** service, in this case **maps**, whose status was checked.
- **status:** string
The status of the checked **TomTom** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.
- **message**
Message to explain what the reported status means.
- **Trimble**
Contains the results of health checks for the respective **Trimble** service.
 - **service:** string
The **Trimble** service, in this case **maps**, whose status was checked.
 - **status:** string
The status of the checked **Trimble** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.
 - **message**
Message to explain what the reported status means.

4.2 Geocoding API

The Geocoding API provides REST services to transform multiple addresses at once into geographical positions using different providers via a unified interface.

With the Geocoding API, customers can choose their service from a list of providers while request and response structures remain unchanged. The addresses can be provided as plain text, and for each address the best-matching result is returned in the form of a GeoJSON feature.

Supported Providers

In the following table, you can find the official API documentation of the respective provider.

Provider	Documentation Link
Here	https://www.here.com/docs/bundle/geocoding-and-search-api-v7-api-reference/page/index.html ➡
Esri	https://developers.arcgis.com/rest/geocode/api-reference/overview-world-geocoding-service.htm ➡
PTV	https://developer.myptv.com/en/documentation/geocoding-places-api/geocoding-places-api-reference ➡
TomTom	https://developer.tomtom.com/geocoding-api/documentation/geocode ➡
Trimble	https://developer.trimblemaps.com/restful-apis/location/single-search/single-search-api/ ➡

📌 Note

Please find a detailed technical API description including Swagger documentation and try-out functionality at https://api.sap.com/api/geocoding_api/overview ➡

Note

When using the Geocoding API relying on the established contract agreements between SAP and **Here**, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Please be aware that this does not generally apply when bringing your own license (BYOL model). In this case, the terms of your own license apply.

Please check the details of the service agreement available at [PROFESSIONAL SERVICES AGREEMENT](#) 

Geocoding API Definition

The Geocoding API provides the following endpoints:

- [POST /geocoding/v1/geocode](#)
- [GET /geocoding/v1/health](#)

Endpoint:

`POST /geocoding/v1/geocode`

Transform a set of addresses into geographical locations in form of GeoJSON. For every provided address, a GeoJSON feature is returned, containing the best-fitting position, its relevance, and address information. Address information is returned even if the provided address was ambiguous (e.g. Leipziger Str., Frankfurt). If a provided address could not be successfully geocoded, an empty GeoJSON feature will be returned.

Parameters (JSON body):

- **credentials**: required (object)

Caution

Starting October 24th, 2023, the two parameters **app_id** and **app_code** for provider **Here** are not supported by SAP HANA spatial services anymore. Use the parameter **api_key**, instead. The reason for this change is that **Here** Technologies deprecated the method of authentication via **app_id** and **app_code**. For more information, see SAP Note [3202728](#) 

The **credentials** object must include a **provider** and an **api_key**.

- **provider**: required (string)
The name of the provider from which to get the geocoding information. Currently, supported values are: **Here**, **Esri**, **PTV**, **TomTom**, **Trimble**.
- **api_key**: required (string)
The API key. If its value is "SAP-KEY", SAP will use established contract agreements with the respective provider for the request. This is currently not supported for providers **Esri**, **PTV**, **TomTom** and **Trimble**, where you will have to bring your own license key.
- **addresses**: required (list)
A comma-separated list of strings. Each string represents an address in freestyle text format.

Please note that the maximum number of addresses to be geocoded in one request depends on several factors such as provider support and license details (in case of BYOL).

Example request:

```
{
  "credentials": {
    "provider": "Here",
    "api_key": "SAP-KEY"
  },
  "addresses": [
    "Bismarckplatz, Heidelberg, Deutschland"
  ]
}
```

Example response:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          8.69304,
          49.40929
        ]
      },
      "properties": {
        "relevance": 1,
        "locationLabel": "Bismarckplatz, 69115 Heidelberg, Deutschland",
        "address": {
          "Label": "Bismarckplatz, 69115 Heidelberg, Deutschland",
          "Country": "DEU",
          "State": "Baden-Württemberg",
          "County": "Heidelberg",
          "City": "Heidelberg",
          "District": "Bergheim",
          "Street": "Bismarckplatz",
          "PostalCode": "69115",
          "AdditionalData": [
            {
              "value": "Deutschland",
              "key": "CountryName"
            },
            {
              "value": "Baden-Württemberg",
              "key": "StateName"
            },
            {
              "value": "Heidelberg",
              "key": "CountyName"
            }
          ]
        },
        "inputIndex": 0,
        "precision": {
          "addressType": "street",
          "matchScore": 1
        }
      }
    }
  ]
}
```

Output values:

The output is a GeoJSON-type FeatureCollection. Each feature has the following structure:

- **geometry**: required (object)
 - **type**: required (string)
Will be of type "Point"
 - **coordinates**: required (list)
Will be empty in case no address could be retrieved for the provided address string.
- **properties**: required (object)
 - **locationLabel**: required (string)
A label for the point of interest. Will be "null" In case no result could be retrieved for the provided address.
 - **relevance**: required (number)
This is the same value as "precision.matchScore" which is the recommended property to retrieve this information from. Please refer to its documentation for a detailed description.
 - **address** : required (object)
A location represented by an address. The combination of fields that are actually filled may vary and depends on the requested provider. Will be "null" In case no result could be retrieved for the provided address.
 - **Label**: string
Assembled address value built out of the parsed address components. In some cases, this might be the only filled field.
 - **City**: string
Input token(s) the parser has categorized as the city part of an address.
 - **Country**: string
ISO 3166-alpha-3 country code
 - **District**: string
Input token(s) the parser has categorized as the district part of an address.
 - **HouseNumber**: string
Input token(s) the parser has categorized as the house number part of an address.
 - **PostalCode**: string
Input token(s) the parser has categorized as the postal code part of an address.
 - **State**: string
Input token(s) the parser has categorized as the state part of an address.
 - **Street**: string
Input token(s) the parser has categorized as the street part of an address (including intersections which are defined by two streets).
 - **AdditionalData**: list of objects
A list of objects, each containing two properties, "key" and "value". The list might be empty, and any content depends on the provider.
 - **inputIndex**: required (number)
Index of the input address from the request's list of addresses. This allows to attribute geocoded results to the address string they have been calculated from. Will be "null" In case no result could be retrieved for the provided address.
 - **precision**: required (object)
Provides information about the quality and precision of the geocoding result.
 - **addressType**: string

The type of address, based on the highest precision the address could be resolved with. If the value is 'other', no information about the type is available. Please note that not all providers support each type of address.

- country
- state
- county
- postalCode
- city
- district
- intersection
- street
- houseNumberInterpolated
- houseNumberExact
- houseNumberWithSupplement
- other
- **matchScore**: number
A normalized value between 0 and 1 indicating how well all tokens from the query address string could be matched to the returned address. This value therefore represents the similarity between the query address string and the determined address. The value 1, for example, indicates that all address tokens could be matched exactly. Note: This is the same value as "relevance", however "matchScore" is the recommended property to use.
For provider **Trimble**, the result is derived from that provider's "ConfidenceLevel". Please check out its definition available at <https://developer.trimblemaps.com/restful-apis/appendix/glossary/> 📖.

Endpoint:

```
GET /geocoding/v1/health
```

The health endpoint can be used to determine the service's current status.

The status code and the response body provide detailed information about the service's health status.

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, esri, ptv, tomtom, trimble

Example response:

```
{
  "status": "UP",
  "timestamp": 1690457952,
  "message": "All services are healthy",
  "details": {
    "HERE": [
      {
        "service": "geocoding",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "ESRI": [
      {
```

```

        "service": "geocoding",
        "status": "OK",
        "message": "Service is healthy"
    },
    "PTV": [
        {
            "service": "geocoding",
            "status": "OK",
            "message": "Service is healthy"
        }
    ],
    "TOMTOM": [
        {
            "service": "geocoding",
            "status": "OK",
            "message": "Service is healthy"
        }
    ],
    "TRIMBLE": [
        {
            "service": "geocoding",
            "status": "OK",
            "message": "Service is healthy"
        }
    ]
}

```

Output values:

- **status:** string
The service's overall health status. The value is "UP", if the service is healthy, and "CRITICAL" otherwise.
- **timestamp:** integer
The timestamp (Unix time) when the service reported the healthy status.
Example: 1606308415
- **message:** string
Message to explain what the reported status means.
Example: All services are healthy
- **details**
An object that contains further details about the checks that lead to the service's reported overall status.
 - **Here**
Contains the results of health checks for the respective **Here** service.
 - **service:** string
The **Here** service, in this case geocoding, whose status was checked (see <https://status.here.com/status>).
 - **status:** string
The status of the checked **Here** service. The value is "OK", if the third party service is reported as healthy, and "CRITICAL" otherwise.
 - **message**
Message to explain what the reported status means.
 - **Esri**
Contains the results of health checks for the respective **Esri** service.
 - **service:** string
The **Esri** service, in this case geocoding, whose status was checked.
 - **status:** string

The status of the checked **Esri** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.

- **message**

Message to explain what the reported status means.

- **PTV**

Contains the results of health checks for the respective **PTV** service.

- **service**: string

The **PTV** service, in this case **geocoding**, whose status was checked.

- **status**: string

The status of the checked **PTV** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.

- **message**

Message to explain what the reported status means.

- **TomTom**

Contains the results of health checks for the respective **TomTom** service.

- **service**: string

The **TomTom** service, in this case **geocoding**, whose status was checked.

- **status**: string

The status of the checked **TomTom** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.

- **message**

Message to explain what the reported status means.

- **Trimble**

Contains the results of health checks for the respective **Trimble** service.

- **service**: string

The **Trimble** service, in this case **geocoding**, whose status was checked.

- **status**: string

The status of the checked **Trimble** service. The value is "OK", if the third-party service is reported as healthy, and "CRITICAL" otherwise.

- **message**

Message to explain what the reported status means.

4.3 Routing APIs

SAP HANA Spatial Services supports both single and matrix routing APIs from different providers via a unified interface.

- The Single Routing API provides services for geographical vehicle routing between one source and one destination location.
- The Matrix Routing API v1 and v2 provide services for geographical vehicle routing between one or many source and destination locations.

4.3.1 Single Routing API

The Single Routing API provides services for geographical vehicle routing between a source and a destination location from different providers via a unified interface.

With the Single Routing API, customers can select their provider of choice from a list of supported providers, without needing to change the request's structure when switching providers. Response structures are also independent of the chosen provider.

Supported Providers

In the following table, you can find the official API documentation of the respective providers.

Provider	Documentation Link
Here	https://www.here.com/docs/category/routing-api-v8 
Esri	https://developers.arcgis.com/rest/routing/ 
PTV	https://developer.myptv.com/documentation/routing-api 
TomTom	https://developer.tomtom.com/routing-api/documentation/tomtom-maps/product-information/introduction 
Trimble	https://developer.trimblemaps.com/restful-apis/routing/route-reports/overview/ 

Note

You can find a detailed technical API description, including Swagger documentation and try-out functionality, at https://api.sap.com/api/routing_api/overview 

Note

When using the Routing API relying on the established contract agreements between SAP and **Here**, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Be aware that this does not generally apply when bringing your own license (BYOL model). In this case, the terms of your own license apply.

To check the details of the service agreement available, see [PROFESSIONAL SERVICES AGREEMENT](#) 

Single Routing API Definitions

The Single Routing API provides the following endpoints:

- [POST /routing/v1/route](#)
- [GET /routing/v1/health](#)

Endpoint:

POST /routing/v1/route

Allows you to get detailed information about a calculated route between a source, a target, and optional waypoint locations. This information includes distance, duration, and route shape.

Parameters (JSON body):

- **credentials:** required (object)
 - **provider:** required (string)
The name of the provider from which to get the routing information. Currently, supported values are: **Here, Esri, PTV, TomTom, Trimble**.
 - **api_key:** required (string)
The API key. If its value is "SAP-KEY", SAP will use established contract agreements with the respective provider for the request. This is currently not supported for providers **Esri, TomTom, and Trimble**, where you must bring your own license key.
 - **providerApiVersion:** optional (string)
The **providerApiVersion** allows you to specify a provider API version to be used for route calculation. This parameter is only supported by provider **Trimble**.
The following values are supported for provider **Trimble**: "PCM34", "PCM35", "PCM36", "PCM37", "PCM38", "PCM_NA", "PCM_EU", "PCM_OC", "PCM_SA", "PCM_ME", "PCM_AS", "PCM_AF", "PCM_GT", "Current".
- **waypoints:** required (GeoJSON MultiPoint)
Specifies the start, the destination, as well as any additional location of the route. At minimum a source (first entry in the list of **coordinates**) and a destination (last entry in the list of **coordinates**) must be provided. If no route could be calculated for a given number of waypoints, an error is returned with status code 400.
- **routeType:** optional (string)
Indicates which type of route is calculated. The default value is **fastest**. The following table provides information about which provider supports which **routeType** values. Note that some parameter combinations such as **vehicleType = truck** and **routeType = shortest** may not be supported by all providers.

Provider	Supported Values	Comments
Here	fastest, shortest	balanced is mapped to fastest
Esri	fastest, shortest	
PTV	fastest, shortest	
TomTom	fastest, shortest	
Trimble	fastest, shortest, balanced	

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **departureTime:** optional (string)
The starting time (UTC) for the route calculation. It could be **now** or any future timestamp. The default value is **now**; that is, the date and time of the request.

- **traffic**: optional (string)
Indicates whether real-time traffic should be used for route calculation. This is set to `true` by default.
- **vehicleType**: optional (string)
Specify the vehicle type that the route should be calculated for. The default value is `truck`. The following table provides information about which provider supports which `vehicleType` values. Note that certain combinations of `vehicleType` and `routeType` may not be supported by all providers (such as `vehicleType = truck` and `routeType = shortest`). Note that **PTV** does not support `vehicleType`; use the `profile` parameter instead.

Provider	Supported Values
Here	truck, car, bicycle, pedestrian
Esri	truck, car, pedestrian
PTV	vehicleType is not supported; use <code>profile</code> instead.
TomTom	truck, car, bicycle, pedestrian
Trimble	truck, car

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **vehicleWidth**: optional (number)
Specify vehicle width, in meters. Must be greater than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleLength**: optional (number)
Specify vehicle length, in meters. Must be greater than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleHeight**: optional (number)
Specify vehicle height, in meters. Must be greater than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleWeight**: optional (number)
Specify vehicle weight including trailers and shipped goods, in kilograms. Must be greater than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **returnGeometry**: optional (Boolean)
Specifies whether to return calculated route as geometry. If set to `false`, it returns an empty linestring geometry. Set to `true` by default.
- **various**: optional (object)
This optional object can contain key-value pairs of provider-specific additional parameters not provided by default or that are vendor specific. May overwrite any already provided or default parameter, but may

not change the general JSON structure of the response. Note that for provider **Trimble**, `HighwayOnly` is switched off by default; it can be set to true by configuring in the `various` parameter accordingly.

- **avoid**: optional (array)
This optional parameter lets you specify a list of categories to be avoided in the route. Currently, the category "tollRoads" is supported. If the category of avoidance is not supported, an error is returned with status code 400.
Example: "avoid": [{"category": "tollRoads"}]
- **profile**: optional (string)
Use a customer or provider-specific profile for routing calculations and benefit from providers' profile capabilities. Only **Trimble** and **PTV** support profile. If set, SAP HANA spatial services only explicitly sets parameters related to response structure and units. This may overwrite the corresponding values if they are configured in the selected profile. SAP HANA spatial services does not implicitly set default values for other parameters, such as "routeType", "traffic", and so on. Note the following differences in behavior for **PTV** and **Trimble**:

Provider	Behavior
PTV	No other parameters of SAP HANA spatial services (except for 'sources', 'destinations', and 'credentials') must be set; otherwise, an error is returned. For example, if the parameter 'routeType' is set to 'shortest', an error is returned. If no profile is set, PTV uses a default value. For a list of supported profiles, see https://developer.myptv.com/en/documentation/matrix-routing-api/concepts/profiles-and-countries . Only capital letters and underscores are allowed in profile value. Note: The behavior in matrix routing is different.
Trimble	All other parameters of SAP HANA spatial services can be set. Any additional parameters provided to profile are passed along to Trimble. Depending on the chosen profile, this can result in profile-specific parameters being overwritten. For a list of provider-defined values, see https://developer.trimblemaps.com/restful-apis/developer-guide/how-to-guides/vehicle-routing-profiles .

Example request:

```
{
  "credentials": {
    "provider": "Here",
    "api_key": "customer_specific_api_key"
  },
  "waypoints": {
    "type": "MultiPoint",
    "coordinates": [
      [
        11.9799793,
        51.4899051
      ],
      [
        11.9797754,
        51.489948
      ],
      [

```

```

        11.9797111,
        51.4900017
      ],
      [
        11.9796789,
        51.4900553
      ]
    ]
  },
  "routeType": "fastest",
  "departureTime": "2023-12-11T15:01:00Z",
  "traffic": true,
  "vehicleType": "truck",
  "vehicleWidth": 2.5,
  "vehicleLength": 5,
  "vehicleHeight": 2,
  "vehicleWeight": 3000
}

```

Example response:

```

{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        11.97998,
        51.4899
      ],
      [
        11.97978,
        51.48995
      ],
      [
        11.97971,
        51.49
      ],
      [
        11.97968,
        51.49006
      ]
    ]
  },
  "properties": {
    "distance": 29,
    "duration": 5,
    "legs": [
      {
        "startPosition": {
          "latitude": 51.4899,
          "longitude": 11.97998
        },
        "endPosition": {
          "latitude": 51.48995,
          "longitude": 11.97978
        },
        "length": 15,
        "duration": 3
      },
      {
        "startPosition": {
          "latitude": 51.48995,
          "longitude": 11.97978
        },
        "endPosition": {

```

```

        "latitude": 51.49,
        "longitude": 11.97971
      },
      "length": 7,
      "duration": 1
    },
    {
      "startPosition": {
        "latitude": 51.49,
        "longitude": 11.97971
      },
      "endPosition": {
        "latitude": 51.49006,
        "longitude": 11.97968
      },
      "length": 7,
      "duration": 1
    }
  ]
}

```

Example request:

```

{
  "credentials": {
    "provider": "Here",
    "api_key": "SAP-KEY"
  },
  "waypoints": {
    "type": "MultiPoint",
    "coordinates": [
      [
        11.9799793,
        51.4899051
      ],
      [
        11.9797754,
        51.489948
      ],
      [
        11.9797111,
        51.4900017
      ],
      [
        11.9796789,
        51.4900553
      ]
    ]
  },
  "routeType": "fastest",
  "departureTime": "2023-12-11T15:01:00Z",
  "traffic": true,
  "vehicleType": "truck",
  "vehicleWidth": 2.5,
  "vehicleLength": 5,
  "vehicleHeight": 2,
  "vehicleWeight": 3000,
  "avoid": [{"category": "tollRoads"}]
}

```

Example response:

```

{
  "type": "Feature",

```

```

    "geometry": {
      "type": "LineString",
      "coordinates": [
        [
          11.97998,
          51.4899
        ],
        [
          11.97978,
          51.48995
        ],
        [
          11.97971,
          51.49
        ],
        [
          11.97968,
          51.49006
        ]
      ]
    },
    "properties": {
      "distance": 29,
      "duration": 5,
      "legs": [
        {
          "startPosition": {
            "latitude": 51.4899,
            "longitude": 11.97998
          },
          "endPosition": {
            "latitude": 51.48995,
            "longitude": 11.97978
          },
          "length": 15,
          "duration": 3
        },
        {
          "startPosition": {
            "latitude": 51.48995,
            "longitude": 11.97978
          },
          "endPosition": {
            "latitude": 51.49,
            "longitude": 11.97971
          },
          "length": 7,
          "duration": 1
        },
        {
          "startPosition": {
            "latitude": 51.49,
            "longitude": 11.97971
          },
          "endPosition": {
            "latitude": 51.49006,
            "longitude": 11.97968
          },
          "length": 7,
          "duration": 1
        }
      ]
    }
  }
}

```

Example request:

```
{
  "credentials": {
    "provider": "PTV",
    "api_key": "customer_specific_api_key"
  },
  "waypoints": {
    "type": "MultiPoint",
    "coordinates": [
      [
        11.9799793,
        51.4899051
      ],
      [
        11.9797754,
        51.489948
      ],
      [
        11.9797111,
        51.4900017
      ],
      [
        11.9796789,
        51.4900553
      ]
    ]
  },
  "profile": "EUR_TRAILER_TRUCK",
  "routeType": "fastest",
  "departureTime": "2020-05-25T15:01:00Z",
  "traffic": true,
  "vehicleWidth": 3,
  "vehicleLength": 5,
  "vehicleHeight": 2,
  "vehicleWeight": 10000
}
```

Example response:

```
{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        11.979966714,
        51.489896083
      ],
      [
        11.979815327,
        51.489976878
      ],
      [
        11.979735579,
        51.490019485
      ],
      [
        11.979676275,
        51.490051151
      ],
      [
        11.979679485,
        51.490055149
      ]
    ]
  }
}
```

```

    ],
    "properties": {
      "distance": 26,
      "duration": 9,
      "legs": [
        {
          "startPosition": {
            "latitude": 51.489896083,
            "longitude": 11.979966714
          },
          "endPosition": {
            "latitude": 51.489976878,
            "longitude": 11.979815327
          },
          "length": 13,
          "duration": 13
        },
        {
          "startPosition": {
            "latitude": 51.489976878,
            "longitude": 11.979815327
          },
          "endPosition": {
            "latitude": 51.490019485,
            "longitude": 11.979735579
          },
          "length": 7,
          "duration": 7
        },
        {
          "startPosition": {
            "latitude": 51.490019485,
            "longitude": 11.979735579
          },
          "endPosition": {
            "latitude": 51.490055149,
            "longitude": 11.979679485
          },
          "length": 6,
          "duration": 6
        }
      ]
    }
  }
}

```

Output values:

The response is a JSON object, containing a GeoJSON LineString. It describes the route by specifying distance and duration for the entire route as well as its legs. By default, it also contains a LineString in GeoJSON format that represents the course of the route.

- **geometry:** required (GeoJSON LineString)
If returnGeometry was set to false, the coordinates is an empty list.
- **properties:** required (number)
 - **distance:** required (integer)
The length of the route in meters.
 - **duration:** required (number)
The duration of the route in seconds.
 - **legs:** required (list of objects)
Each object in the list contains the following:
 - **startPosition:** required (object)

- latitude
- longitude
- **endPosition**: required (object)
 - latitude
 - longitude
- **length**: required (integer)
Length of the leg in meters.
- **duration**: required (integer)
Duration of the leg in seconds.

Endpoint:

```
GET /routing/v1/health
```

The health endpoint can be used to determine the service's current status.

The status code and the response body provide detailed information about the service's health status.

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, esri, ptv, tomtom, trimble

Example response:

```
{
  "status": "UP",
  "timestamp": 1690457989,
  "message": "All services are healthy",
  "details": {
    "HERE": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "ESRI": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "PTV": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "TOMTOM": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "TRIMBLE": [
      {
```

```

    "service": "routing",
    "status": "OK",
    "message": "Service is healthy"
  }
]
}

```

Output values:

- **status:** string
The service's overall health status. The value is "UP" if the service is healthy; otherwise, "CRITICAL".
- **timestamp:** integer
The timestamp (Unix time) when the service reported the healthy status.
Example: 1606308415
- **message:** string
Message that explains what the reported status means.
Example: All services are healthy
- **details**
An object that contains further details about the checks that lead to the service's reported overall status.
 - **Here**
Contains the results of health checks for the respective **Here** service.
 - **service:** string
The **Here** service, in this case `routing`, whose status was checked (see <https://status.here.com/status>).
 - **status:** string
The status of the checked **Here** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means.
 - **Esri**
Contains the results of health checks for the respective **Esri** service.
 - **service:** string
The **Esri** service, in this case `routing`, whose status was checked.
 - **status:** string
The status of the checked **Esri** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means.
 - **PTV**
Contains the results of health checks for the respective **PTV** service.
 - **service:** string
The **PTV** service, in this case `routing`, whose status was checked.
 - **status:** string
The status of the checked **PTV** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means.
 - **TomTom**

Contains the results of health checks for the respective **TomTom** service.

- **service:** string
The **TomTom** service, in this case **routing**, whose status was checked.
- **status:** string
The status of the checked **TomTom** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
- **message**
Message that explains what the reported status means.
- **Trimble**
Contains the results of health checks for the respective **Trimble** service.
 - **service:** string
The **Trimble** service, in this case **routing**, whose status was checked.
 - **status:** string
The status of the checked **Trimble** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means.

4.3.2 Matrix Routing API

The Matrix Routing API v1 and v2 provide services for geographical vehicle routing between one or many source and destination locations from different providers via a unified interface.

With the Matrix Routing API, customers can select their provider of choice from a list of supported providers, without needing to change the request's structure when switching providers. Response structures are also independent of the chosen provider.

Supported Providers

In the following table, you can find the official API documentation of the respective providers.

Provider	Documentation Link
Here	https://www.here.com/docs/category/routing-api-v8 ➡
Esri	https://developers.arcgis.com/rest/routing/ ➡
PTV	https://developer.myptv.com/documentation/routing-api ➡
TomTom	https://developer.tomtom.com/routing-api/documentation/tomtom-maps/product-information/introduction ➡
Trimble	https://developer.trimblemaps.com/restful-apis/routing/route-reports/overview/ ➡

Differences between Matrix Routing v1 and Matrix Routing v2

- Matrix Routing v1 has been deprecated, and it offers **HERE** as the provider for the OEM licensing model.
- Matrix Routing v2 offers **TomTom** as the provider for the OEM licensing model.

Note

You can find a detailed technical API description including Swagger documentation and try-out functionality at https://api.sap.com/api/routing_api/overview.

Note

Matrix Routing v1:

When using the Routing API relying on the established contract agreements between SAP and Here, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Matrix Routing v2:

When using the Routing API relying on the established contract agreements between SAP and TomTom, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Be aware that this does not generally apply when bringing your own license (BYOL model). In this case, the terms of your own license apply.

Check the details of the service agreement available at [PROFESSIONAL SERVICES AGREEMENT](#).

Caution

Matrix Routing v1 has been deprecated on 1 September 2025, and will sunset after 12 months. For more information, see SAP Note [3648907](#).

Matrix Routing API Definitions

The Matrix Routing API provides the following endpoints:

- [POST /routing/v1/matrix](#)
- [POST /routing/v2/matrix](#)
- [GET /routing/v1/health](#)
- [GET /routing/v2/health](#)

Endpoint:

`POST /routing/v1/matrix`

Allows you to get information about routes between large sets of source and destination coordinates. Results include distances and travel durations calculated for all possible combinations of given sources and destinations.

Parameters (JSON body):

- **credentials:** required (object)
The **credentials** object must include a **provider** and an **api_key**.
 - **provider:** required (string)
The name of the provider from which to get the routing information. Currently, supported values are: **Here**, **Esri**, **PTV**, **TomTom**, **Trimble**.
 - **api_key:** required (string)
The API key. If its value is "SAP-KEY", SAP uses established contract agreements with the respective provider for the request. For the current version of Matrix Routing (v1), this is only supported for **HERE**. It is not currently supported for providers **Esri**, **PTV**, **TomTom**, and **Trimble**, for which you must use your own license key.
- **sources:** required (GeoJSON MultiPoint)
A list of up to 100 sources to calculate routes from. **PTV** supports a maximum of 500 sources. If a provider supports less than 100 sources, SAP HANA spatial services internally creates sub-matrices to accommodate for the provider's limits. For more information about provider-specific limitations, check the official API documentation of the respective provider.

Provider	Maximum Number of Sources
Here	100
Esri	100
PTV	500
TomTom	100
Trimble	100

- **destinations:** required (GeoJSON MultiPoint)
A list of up to 100 destinations to calculate routes to. **PTV** supports a maximum of 500 destinations. If a provider supports less than 100 destinations, SAP HANA spatial services internally creates sub-matrices to accommodate for the provider's limits. For more information about provider-specific limitations, check the official API documentation of the respective provider.

Provider	Maximum Number of Destinations
Here	100
Esri	100
PTV	500
TomTom	100
Trimble	100

- **routeType:** optional (string)
Indicates which type of route is calculated. The default value is **fastest**. The following table provides information about which provider supports which **routeType** values. Note that some parameter combinations such as **vehicleType** = **truck** and **routeType** = **shortest** may not be supported by all providers. Note that **PTV** does not support **routeType**.

Provider	Supported Values	Comments
Here	fastest, shortest	For compatibility reasons, balanced is still supported and is mapped to fastest .
Esri	fastest, shortest	
PTV		routeType is not supported.
TomTom	fastest	
Trimble	fastest, shortest, balanced	

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **departureTime**: optional (string)
The starting time (UTC) for the route calculation. It could be **now** or any future timestamp. The default value is **now**; that is, the date and time of the request. Note that **PTV** does not support **departureTime**.
- **traffic**: optional (Boolean)
Indicates whether real-time traffic should be used for route calculation. This is set to **true** by default. Note that **PTV** does not support **traffic=true**.
- **vehicleType**: optional (string)
Specify the vehicle type that the route matrix should be calculated for. The default value is **truck**. The following table provides information about which provider supports which **vehicleType** values. Note that certain combinations of **vehicleType** and **routeType** may not be supported by all providers (such as **vehicleType** = **truck** and **routeType** = **shortest**). Note that **PTV** does not support **vehicleType**; use the **profile** parameter instead.

Provider	Supported Values
Here	truck, car, bicycle, pedestrian
Esri	truck, car, pedestrian
PTV	vehicleType is not supported; use profile instead
TomTom	truck, car, pedestrian
Trimble	truck, car

- **vehicleWidth**: optional (number)
Specify vehicle width, in meters. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleType**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleLength**: optional (number)
Specify vehicle length, in meters. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleLength**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **vehicleHeight:** optional (number)
Specify vehicle height, in meters. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleHeight**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleWeight:** optional (number)
Specify vehicle weight including trailers and shipped goods, in kilograms. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleWeight**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **various:** optional (object)
This optional object can contain key-value pairs of provider-specific additional parameters not provided by default or that are vendor specific. It may overwrite any already provided or default parameter, but may not change the general JSON structure of the response. Note that for provider **Trimble**, `HighwayOnly` is switched off by default; it can be set to true by configuring in the `various` parameter accordingly. Note that **PTV** does not support **various**.
- **avoid:** optional (array)
This optional parameter allows to specify a list of categories to be avoided in the route. Currently, the category "tollRoads" is supported. If the category of **avoid** is not supported, an error is returned with status code 400. Note that **PTV** does not support **avoid**. For more information about provider-specific limitations, check the official API documentation of the respective provider.
Example: "avoid": [{"category": "tollRoads"}]
- **profile:** optional (string)
Use a customer or provider-specific profile for routing calculations and benefit from providers' profile capabilities. Only **Trimble** and **PTV** support profile. If set, SAP HANA spatial services will only explicitly set parameters related to response structure and units. This may overwrite the corresponding values in case they are configured in the selected profile. HANA spatial services won't implicitly set any default values for other parameters, such as "routeType", "traffic", and so on. Note the following differences in behavior for **PTV** and **Trimble**:

Provider	Behavior
PTV	No other parameters of SAP HANA spatial services (except for 'sources', 'destinations' and 'credentials') must be set. An error will be returned otherwise. For example, if the parameter 'routeType' is set to 'shortest' an error will be returned. If no profile is set, PTV will use a default value. For a list of supported profiles, see https://developer.mypTV.com/en/documentation/matrix-routing-api/concepts/profiles-and-countries . Only capital letters and underscores are allowed in profile value. Note: The behavior in single routing is different.

Provider	Behavior
Trimble	All other parameters of SAP HANA spatial services can be set. Any parameter provided additionally to profile will be passed along to Trimble. Depending on the chosen profile, this can result in profile-specific parameters being overwritten. For a list of supported profiles, see https://developer.trimblemaps.com/restful-apis/developer-guide/how-to-guides/vehicle-routing-profiles  .

For more information about provider-specific limitations, check the official API documentation of the respective provider.

Example request:

```
{
  "credentials": {
    "provider": "Here",
    "api_key": "customer_specific_api_key"
  },
  "sources": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.4,
        52.5
      ],
      [
        13.4,
        52.5
      ]
    ]
  },
  "destinations": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.43,
        52.5
      ],
      [
        13.46,
        52.5
      ]
    ]
  },
  "routeType": "fastest",
  "departureTime": "2020-05-25T15:01:00Z",
  "traffic": true,
  "vehicleType": "truck",
  "vehicleWidth": 3,
  "vehicleLength": 5,
  "vehicleHeight": 2,
  "vehicleWeight": 10000,
  "various": {}
}
```

Example response:

```
[
  {
    "startIndex": 0,
    "destinationIndex": 0,
    "summary": {
```

```

        "distance": 372,
        "travelTime": 26
    },
    {
        "startIndex": 0,
        "destinationIndex": 1,
        "status": "failed"
    },
    {
        "startIndex": 1,
        "destinationIndex": 0,
        "summary": {
            "distance": 434,
            "travelTime": 30
        }
    },
    {
        "startIndex": 1,
        "destinationIndex": 1,
        "status": "failed"
    }
]

```

Example request:

```

{
  "credentials": {
    "provider": "Here",
    "api_key": "SAP-KEY"
  },
  "sources": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.4,
        52.5
      ],
      [
        13.4,
        52.5
      ]
    ]
  },
  "destinations": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.43,
        52.5
      ],
      [
        13.46,
        52.5
      ]
    ]
  },
  "routeType": "fastest",
  "departureTime": "2023-12-11T15:01:00Z",
  "traffic": true,
  "vehicleType": "truck",
  "vehicleWidth": 3,
  "vehicleLength": 5,
  "vehicleHeight": 2,
  "vehicleWeight": 10000,
  "various": {},
  "avoid": [{"category": "tollRoads"}]
}

```

```
}
```

Example response:

```
[
  {
    "startIndex": 0,
    "destinationIndex": 0,
    "summary": {
      "distance": 2433,
      "travelTime": 576
    }
  },
  {
    "startIndex": 0,
    "destinationIndex": 1,
    "summary": {
      "distance": 9508,
      "travelTime": 1844
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 0,
    "summary": {
      "distance": 2433,
      "travelTime": 576
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 1,
    "summary": {
      "distance": 9508,
      "travelTime": 1844
    }
  }
]
```

Example request:

```
{
  "credentials": {
    "provider": "PTV",
    "api_key": "customer_specific_api_key"
  },
  "profile": "EUR_TRAILER_TRUCK",
  "sources": {
    "type": "MultiPoint",
    "coordinates": [
      [13.4, 52.5],
      [13.4, 52.5]
    ]
  },
  "destinations": {
    "type": "MultiPoint",
    "coordinates": [
      [13.43, 52.5],
      [13.46, 52.5]
    ]
  }
}
```

Example response:

```
[
  {
    "startIndex": 0,
    "destinationIndex": 0,
    "summary": {
      "distance": 2999,
      "travelTime": 581
    }
  },
  {
    "startIndex": 0,
    "destinationIndex": 1,
    "summary": {
      "distance": 6410,
      "travelTime": 1101
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 0,
    "summary": {
      "distance": 2999,
      "travelTime": 581
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 1,
    "summary": {
      "distance": 6410,
      "travelTime": 1101
    }
  }
]
```

Output values:

The response is a list of JSON objects with the following structure:

- **startIndex**: required (integer)
Index of the coordinate pair in the request's "sources" multipoint geometry that is used as starting point (source) of the route.
- **destinationIndex**: required (integer)
Index of the coordinate pair in the request's "destinations" multipoint geometry that is used as endpoint (destination) of the route.

In addition, the response can contain one of the following two properties (summary or status):

- **summary**: required (object):
 - **distance**: required (integer)
The total travel distance of the route in meters.
 - **travelTime**: required (integer)
The total travel time for the route in seconds.
- **status**: required (string)
Status text indicating that no route could be calculated for a defined combination of start and end point. This text can include `failed` or `rangeExceeded`. This field is not present if the route calculation succeeded.

Endpoint:

POST /routing/v2/matrix

Allows you to get information about routes between large sets of source and destination coordinates. Results include distances and travel durations calculated for all possible combinations of given sources and destinations.

Parameters (JSON body):

- **credentials:** required (object)
The **credentials** object must include a **provider** and an **api_key**.
 - **provider:** required (string)
The name of the provider from which to get the routing information. Currently, supported values are: **Here**, **Esri**, **PTV**, **TomTom**, **Trimble**.
 - **api_key:** required (string)
The API key. If its value is "SAP-KEY", SAP uses established contract agreements with the respective provider for the request. For the current version of Matrix Routing (v2), this is only supported for **TomTom**. It is not currently supported for providers **Esri**, **HERE**, **PTV**, and **Trimble**, for which you must bring your own license key.
- **sources:** required (GeoJSON MultiPoint)
A list of up to 100 sources to calculate routes from. **PTV** supports a maximum of 500 sources. If a provider supports less than 100 sources, SAP HANA spatial services internally creates sub-matrices to accommodate for the provider's limits.

Provider	Maximum Number of Sources
Here	100
Esri	100
PTV	500
TomTom	100
Trimble	100

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **destinations:** required (GeoJSON MultiPoint)
A list of up to 100 destinations to calculate routes to. **PTV** supports a maximum of 500 destinations. If a provider supports less than 100 destinations, SAP HANA spatial services internally creates sub-matrices to accommodate for the provider's limits.

Provider	Maximum Number of Destinations
Here	100
Esri	100
PTV	500
TomTom	100
Trimble	100

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **routeType:** optional (string)
Indicates which type of route is calculated. The default value is **fastest**. The following table provides information about which provider supports which **routeType** values. Note that some parameter combinations such as **vehicleType = truck** and **routeType = shortest** may not be supported by all providers. Note that **PTV** does not support **routeType**.

Provider	Supported Values	Comments
Here	fastest, shortest	For compatibility reasons, balanced is still supported and is mapped to fastest .
Esri	fastest, shortest	
PTV		routeType is not supported.
TomTom	fastest	
Trimble	fastest, shortest, balanced	

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **departureTime:** optional (string)
The starting time (UTC) for the route calculation. It could be **now** or any future timestamp. The default value is **now**; that is, the date and time of the request. Note that **PTV** does not support **departureTime**.
- **traffic:** optional (Boolean)
Indicates whether real time traffic should be used for route calculation. This is set to **true** by default. Note that **PTV** does not support **traffic=true**.
- **vehicleType:** optional (string)
Specify the vehicle type that the route matrix should be calculated for. The default value is **truck**. The following table provides information about which provider supports which **vehicleType** values. Note that certain combinations of **vehicleType** and **routeType** may not be supported by all providers (such as **vehicleType = truck** and **routeType = shortest**). Note that **PTV** does not support **vehicleType**; use the **profile** parameter instead.

Provider	Supported Values
Here	truck, car, bicycle, pedestrian
Esri	truck, car, pedestrian
PTV	vehicleType is not supported; use profile instead
TomTom	truck, car, pedestrian
Trimble	truck, car

For more information about provider-specific limitations, check the official API documentation of the respective provider.

- **vehicleWidth:** optional (number)
Specify vehicle width, in meters. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleType**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleLength:** optional (number)
Specify vehicle length, in meters. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleLength**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleHeight:** optional (number)
Specify vehicle height, in meters. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleHeight**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **vehicleWeight:** optional (number)
Specify vehicle weight including trailers and shipped goods, in kilograms. Must be larger than 0. Provider-specific upper limits may apply. Depending on the chosen provider, this parameter may be ignored in combination with certain vehicle types. For example, **Here** only supports this parameter with truck routing. Note that **PTV** does not support **vehicleWeight**; all vehicle attributes are defined as part of their default profiles. For more information about provider-specific limitations, check the official API documentation of the respective provider.
- **various:** optional (object)
This optional object can contain key-value pairs of provider-specific additional parameters not provided by default or that are vendor specific. It may overwrite any already provided or default parameter, but may not change the general JSON structure of the response. Note that for provider **Trimble**, ``HighwayOnly`` is switched off by default; it can be set to true by configuring the ``various`` parameter accordingly. Note that **PTV** does not support **various**.
- **avoid:** optional (array)
This optional parameter lets you specify a list of categories to be avoided in the route. Currently, the category "tollRoads" is supported. If the category of **avoid** is not supported, an error is returned with the status code 400. Note that **PTV** does not support **avoid**. For more information about provider-specific limitations, check the official API documentation of the respective provider.
Example: `"avoid": [{"category": "tollRoads"}]`
- **profile:** optional (string)
Use a customer or provider-specific profile for routing calculations and benefit from providers' profile capabilities. Only **Trimble** and **PTV** support profile. If set, SAP HANA spatial services only explicitly sets parameters related to response structure and units. This may overwrite the corresponding values if they are configured in the selected profile. SAP HANA spatial services will not implicitly set any default values

for other parameters, such as "routeType", "traffic", and so on. Note the following differences in behavior for **PTV** and **Trimble**:

Provider	Behavior
PTV	No other parameters of SAP HANA spatial services (except for 'sources', 'destinations', and 'credentials') must be set; otherwise, an error is returned. For example, if the parameter 'routeType' is set to 'shortest' an error is returned. If no profile is set, PTV uses a default value. For a list of supported profiles, see https://developer.myptv.com/en/documentation/matrix-routing-api/concepts/profiles-and-countries . Only capital letters and underscores are allowed in profile value. Note: The behavior is different in single routing.
Trimble	All other parameters of SAP HANA spatial services can be set. Any parameter provided additionally to profile are passed along to Trimble. Depending on the chosen profile, this can result in profile-specific parameters being overwritten. For a list of provider-defined values, see https://developer.trimblemaps.com/restful-apis/developer-guide/how-to-guides/vehicle-routing-profiles .

Example request:

```
{
  "credentials": {
    "provider": "TomTom",
    "api_key": "customer_specific_api_key"
  },
  "sources": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.4,
        52.5
      ],
      [
        13.4,
        52.5
      ]
    ]
  },
  "destinations": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.43,
        52.5
      ],
      [
        13.46,
        52.5
      ]
    ]
  },
  "routeType": "fastest",
  "traffic": true,
  "vehicleType": "truck",
}
```

```

"vehicleWidth": 3,
"vehicleLength": 5,
"vehicleHeight": 2,
"vehicleWeight": 10000,
"various": {}
}

```

Example response:

```

[
  {
    "startIndex": 0,
    "destinationIndex": 0,
    "summary": {
      "distance": 372,
      "travelTime": 26
    }
  },
  {
    "startIndex": 0,
    "destinationIndex": 1,
    "status": "failed"
  },
  {
    "startIndex": 1,
    "destinationIndex": 0,
    "summary": {
      "distance": 434,
      "travelTime": 30
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 1,
    "status": "failed"
  }
]

```

Example request:

```

{
  "credentials": {
    "provider": "TomTom",
    "api_key": "SAP-KEY"
  },
  "sources": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.4,
        52.5
      ],
      [
        13.4,
        52.5
      ]
    ]
  },
  "destinations": {
    "type": "MultiPoint",
    "coordinates": [
      [
        13.43,
        52.5
      ],
      [

```

```

        13.46,
        52.5
    ]
  ],
  "routeType": "fastest",
  "traffic": true,
  "vehicleType": "truck",
  "vehicleWidth": 3,
  "vehicleLength": 5,
  "vehicleHeight": 2,
  "vehicleWeight": 10000,
  "various": {},
  "avoid": [{"category": "tollRoads"}]
}

```

Example response:

```

[
  {
    "startIndex": 0,
    "destinationIndex": 0,
    "summary": {
      "distance": 2433,
      "travelTime": 576
    }
  },
  {
    "startIndex": 0,
    "destinationIndex": 1,
    "summary": {
      "distance": 9508,
      "travelTime": 1844
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 0,
    "summary": {
      "distance": 2433,
      "travelTime": 576
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 1,
    "summary": {
      "distance": 9508,
      "travelTime": 1844
    }
  }
]

```

Example request:

```

{
  "credentials": {
    "provider": "PTV",
    "api_key": "customer_specific_api_key"
  },
  "profile": "EUR_TRAILER_TRUCK",
  "sources": {
    "type": "MultiPoint",
    "coordinates": [
      [13.4, 52.5],
      [13.4, 52.5]
    ]
  }
}

```

```

    ]
  },
  "destinations": {
    "type": "MultiPoint",
    "coordinates": [
      [13.43, 52.5],
      [13.46, 52.5]
    ]
  }
}

```

Example response:

```

[
  {
    "startIndex": 0,
    "destinationIndex": 0,
    "summary": {
      "distance": 2999,
      "travelTime": 581
    }
  },
  {
    "startIndex": 0,
    "destinationIndex": 1,
    "summary": {
      "distance": 6410,
      "travelTime": 1101
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 0,
    "summary": {
      "distance": 2999,
      "travelTime": 581
    }
  },
  {
    "startIndex": 1,
    "destinationIndex": 1,
    "summary": {
      "distance": 6410,
      "travelTime": 1101
    }
  }
]

```

Output values:

The response is a list of JSON objects with the following structure:

- **startIndex:** required (integer)
Index of the coordinate pair in the request's "sources" multipoint geometry that is used as starting point (source) of the route.
- **destinationIndex:** required (integer)
Index of the coordinate pair in the request's "destinations" multipoint geometry that is used as endpoint (destination) of the route.

In addition, the response can contain one of the following two properties (summary or status):

- **summary:** required (object):
 - **distance:** required (integer)

The total travel distance of the route in meters.

- **travelTime**: required (integer)
The total travel time for the route in seconds.
- **status**: required (string)
Status text indicating that no route could be calculated for a defined combination of start and endpoint. This text can include `failed` or `rangeExceeded`. This field is not present if the route calculation succeeded.

Endpoint:

```
GET /routing/v1/health
GET /routing/v2/health
```

The health endpoints can be used to determine the service's current status.

The status code and the response body provide detailed information about the service's health status.

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, esri, ptv, tomtom, trimble

Example response:

```
{
  "status": "UP",
  "timestamp": 1690457989,
  "message": "All services are healthy",
  "details": {
    "HERE": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "ESRI": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "PTV": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "TOMTOM": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ],
    "TRIMBLE": [
      {
        "service": "routing",
        "status": "OK",
        "message": "Service is healthy"
      }
    ]
  }
}
```

```

    }
  ]
}

```

Output values:

- **status:** string
The service's overall health status. The value is "UP" if the service is healthy; otherwise, "CRITICAL" .
- **timestamp:** integer
The timestamp (Unix time) when the service reported the healthy status.
Example: 1606308415
- **message:** string
Message that explains what the reported status means.
Example: All services are healthy
- **details**
An object that contains further details about the checks that lead to the service's reported overall status.
 - **Here**
Contains the results of health checks for the respective **Here** service.
 - **service:** string
The **Here** service, in this case `routing`, whose status was checked (see <https://status.here.com/status>).
 - **status:** string
The status of the checked **Here** service. The value is "OK" if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means.
 - **Esri**
Contains the results of health checks for the respective **Esri** service.
 - **service:** string
The **Esri** service, in this case `routing`, whose status was checked.
 - **status:** string
The status of the checked **Esri** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means. what the reported status means.
 - **PTV**
Contains the results of health checks for the respective **Esri** service.
 - **service:** string
The **PTV** service, in this case `routing`, whose status was checked.
 - **status:** string
The status of the checked **PTV** service. The value is "OK", if the third-party service is reported as healthy; otherwise, "CRITICAL".
 - **message**
Message that explains what the reported status means.
 - **TomTom**
Contains the results of health checks for the respective **TomTom** service.
 - **service:** string

The **TomTom** service, in this case `routing`, whose status was checked.

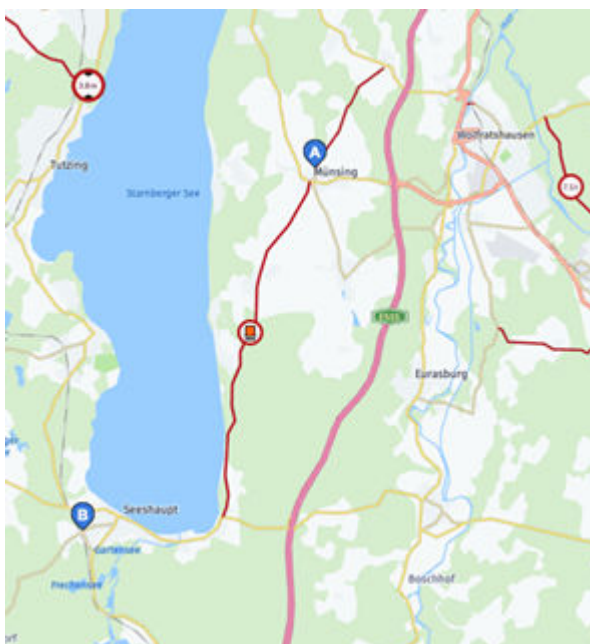
- **status**: string
The status of the checked **TomTom** service. The value is "OK", if the third-party service is reported as healthy; otherwise "CRITICAL".
- **message**
Message that explains what the reported status means.
- **Trimble**
Contains the results of health checks for the respective **Trimble** service.
 - **service**: string
The **Trimble** service, in this case `routing`, whose status was checked.
 - **status**: string
The status of the checked **Trimble** service. The value is "OK", if the third-party service is reported as healthy; otherwise "CRITICAL".
 - **message**
Message that explains what the reported status means.

4.3.3 Using the `various` Parameter

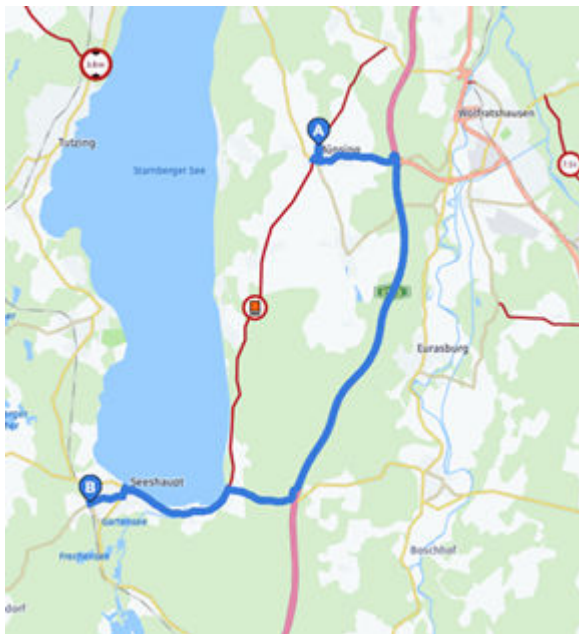
The following examples show how requests to the SAP HANA spatial services Single Routing and Matrix Routing APIs can be configured by specifying provider-specific request parameters, which aren't part of the [Routing API definitions](#) of SAP HANA spatial services.

Example: Apply Truck Restrictions

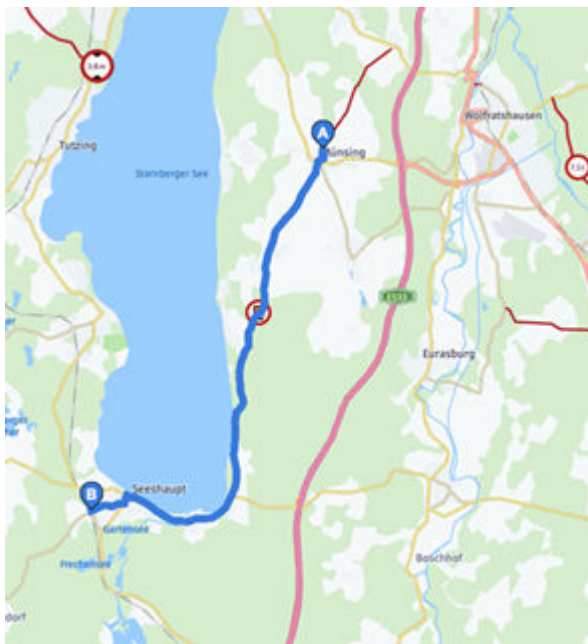
A truck must drive from the city of "Münsing" (see pin "A" picture below) to "Seeshaupt" (see pin "B" on picture below). The truck is loaded with hazardous material that is considered harmful to water. The direct connection between both cities takes a route that has a restriction for trucks with such a load:



Therefore, on the way from A to B, the truck must take a slight detour to avoid taking a restricted road:



Having delivered its load, on the way back such restrictions don't apply to the now-empty truck, which is the way it can take the previously restricted and shorter route:



How to achieve this with provider HERE:

```
{
  "credentials": {
    "provider": "here",
    "api_key": "SAP-KEY"
  },
  "waypoints": {
    "type": "MultiPoint",
```

```

    "coordinates": [
      [
        11.369605591718596,
        47.906146850882635
      ],
      [
        11.288877426018132,
        47.821738268453686
      ]
    ]
  },
  "routeType": "fastest",
  "traffic": true,
  "vehicleType": "truck",
  "various": {
    "vehicle[shippedHazardousGoods]": "harmfulToWater"
  }
}

```

How to achieve this with provider TomTom:

```

{
  "credentials": {
    "provider": "tomtom",
    "api_key": "{{ _['TOMTOM-API-KEY'] }}"
  },
  "waypoints": {
    "type": "MultiPoint",
    "coordinates": [
      [
        11.369605591718596,
        47.906146850882635
      ],
      [
        11.288877426018132,
        47.821738268453686
      ]
    ]
  },
  "routeType": "fastest",
  "traffic": true,
  "vehicleType": "truck",
  "various": {
    "vehicleLoadType": "otherHazmatHarmfulToWater"
  }
}

```

How to achieve this with provider Trimble:

```

{
  "credentials": {
    "provider": "trimble",
    "api_key": "{{ _['TRIMBLE-API-KEY'] }}"
  },
  "waypoints": {
    "type": "MultiPoint",
    "coordinates": [
      [
        -122.25060339668153,
        37.821379240983035
      ],
      [
        -122.42853328803193,
        37.77551677944018
      ]
    ]
  }
}

```

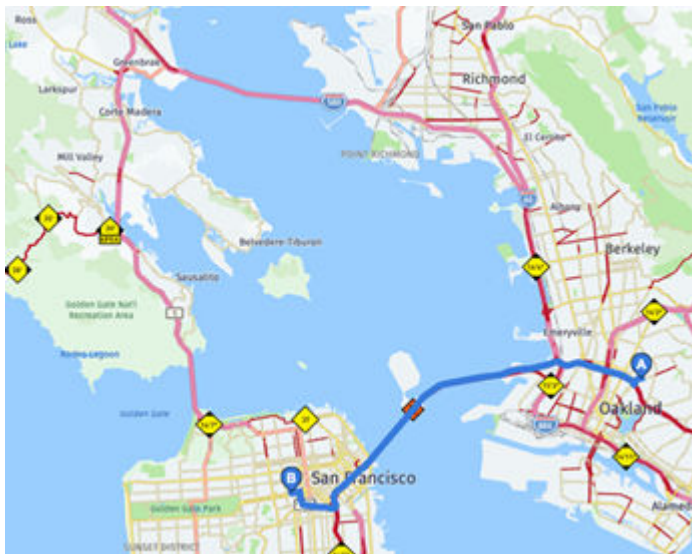
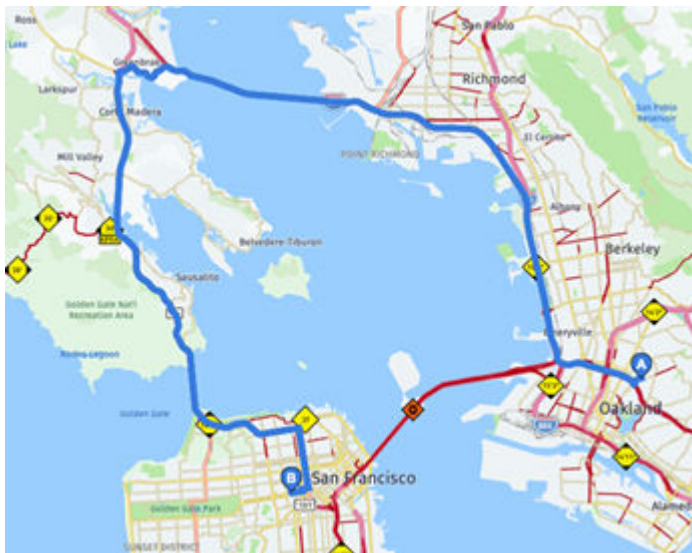
```

    },
    "routeType": "fastest",
    "traffic": false,
    "vehicleType": "car",
    "various": {
      "Options": {
        "HazMatTypes": [
          3
        ]
      }
    }
  }
}

```

Note

For Trimble, a location in the USA was chosen.



Different Examples for Different Providers

The provider-specific parameters can also be combined using the "various" property.

HERE

Description	Code
A Truck loads hazardous material that is "harmful to water"	<pre>"various": { "vehicle[shippedHazardousGoods]": "harmfulToWater" }</pre>
A Truck is only allowed to travel at 60 km/h (HERE requires the unit to be m/s)	<pre>"various": { "vehicle[speedCap]": 16.667 }</pre>
Avoid a certain area as defined via a bounding box (lower left corner, upper right corner)	<pre>"various": { "avoid[areas]": "bbox:13.082, 52.416, 13.628, 52.626" }</pre>

TomTom

Description	Code
A Truck loads hazardous material that is "harmful to water"	<pre>"various": { "vehicleLoadType": "otherHazmatHarmfulToWater" }</pre>
A Truck is only allowed to travel at 60 km/h	<pre>"various": { "vehicleMaxSpeed": 60 }</pre>

Description	Code
Avoid a certain area as defined via a bounding box (lower left corner, upper right corner)	⚠ Caution TomTom's parameter "avoidAreas" isn't yet supported by HSS via "various". It's listed here to indicate how HSS' "various" parameter supports nested JSON objects mixed with different types. <pre>"various": { "avoidAreas": { "rectangles": [{ "southWestCorner": { "latitude": 47.82063804415921, "longitude": 11.333597414431154 }, "northEastCorner": { "latitude": 47.82559856527993, "longitude": 11.343448196565362 } }] } }</pre>

Trimble

Description	Code
A Truck loads hazardous material that is "harmful to water"	<pre>"various": { "Options": { "HazMatTypes" : [3] } }</pre>
A Truck is only allowed to travel at 60 km/h	<pre>"various": { "Options": { "GovernorSpeedLimit": "60" } }</pre>
Avoid a certain area as defined via a bounding box (lower left corner, upper right corner)	Not supported by Trimble

PTV

Description	Code
A Truck loads hazardous material that is "harmful to water"	<pre>"various": { "vehicle[hazardousMaterials]": "HAZARDOUS_TO_WATER" }</pre>

Description	Code
A Truck is only allowed to travel at 60 km/h	<pre>"various": { "options[maximumSpeed]": 60 }</pre>
Avoid a certain area as defined via a bounding box (lower left corner, upper right corner)	Not supported by PTV

4.4 API Transaction Costs

The individual SAP HANA spatial services APIs differ in their complexity involved in performing the respective calculations. Therefore, single requests are metered and billed differently.

API	Details
Mapping API	Depending on the path, a request will be equivalent to different amounts of transactions. Detailed conversion between requests and transactions [page 66]
Routing API	Depending on the path and the presence of certain parameters, a request will be equivalent to different amounts of transactions. Detailed conversion between requests and transactions [page 66]
Geocoding API	- Using SAP Global OEM Contract with HERE ("SAP-KEY" credential): 200 Addresses to be Geocoded = 1 Transaction (1/200 transactions per address) - Using own credentials for the 3rd party provider: 10,000 Addresses to be Geocoded = 1 Transaction (1/10,000 transactions per address)

Detailed conversion between requests and transactions:

API	Method / Path / Parameters	Costs
Mapping API	/here/tiles	3000 Requests = 1 Transaction (1/3000 transactions per request) (1 map view usually consists of 15 requested map tiles, which is calculated as 1/200 transactions)

API	Method / Path / Parameters	Costs
Routing API, /route endpoint	vehicleType=car vehicleType=bicycle vehicleType=pedestrian	- Using SAP Global OEM Contract with HERE ("SAP-KEY" credential): 120 Requests = 1 Transaction (1/120 transactions per request) - Using own credentials for the 3rd party provider: 10,000 Requests = 1 Transaction (1/10,000 transactions per request)
	vehicleType=truck	- Using SAP Global OEM Contract with HERE ("SAP-KEY" credential): 70 Requests = 1 Transaction (1/70 transactions per request) - Using own credentials for the 3rd party provider: 10,000 Requests = 1 Transaction (1/10,000 transactions per request)
Routing API, /matrix endpoint	vehicleType=car	- Using SAP Global OEM Contract with HERE ("SAP-KEY" credential): 120 Starting Locations = 1 Transaction (*) - Using own credentials for the 3rd party provider: 10,000 Requests = 1 Transaction (1/10,000 transactions per request)
	vehicleType=truck	- Using SAP Global OEM Contract with HERE ("SAP-KEY" credential): 70 Starting Locations = 1 Transaction (*) - Using own credentials for the 3rd party provider: 10,000 Requests = 1 Transaction (1/10,000 transactions per request)

(*): For matrix routing, the number of starting locations also determines the costs. For example, 1 matrix routing request for vehicleType=truck with 12 starting locations (parameter sources), and 25 destination locations (parameter destinations) will be equivalent to $(12 \cdot 1) / 70 = 12/70$ transaction.

Note

For further details on transaction billing and metering, see the API descriptions on SAP BTP Marketplace as well as the SAP HANA Spatial Services service description available at [PROFESSIONAL SERVICES AGREEMENT](#).

4.5 API Responses

The following describes the response codes that can result from any call of the spatial services APIs.

Response Code	Description
200	The request was evaluated successfully.
400	Bad request (header data, parameter, or request body wrong or incomplete).
401	The given request is unauthorized.
403	The evaluated request scopes in the access token are not sufficient and do not match the required scopes.
500	An internal server error occurred.

Further details clarifying the service responses upon successful evaluation can be found within the response body.

5 Frequently Asked Questions for SAP HANA Spatial Services

This page addresses questions that have been frequently asked about SAP HANA spatial services. The questions are organized into four categories: **General information**, **Technical Features and Configuration**, **License and Pricing**, and **Integration and Availability**.

Contents

General Information

1. What is SAP HANA spatial services and what does it offer? [page 70]
2. How does SAP HANA spatial services enhance spatial data processing and analysis? [page 70]
3. What are the key features of SAP HANA spatial services? [page 70]
4. How can SAP HANA spatial services benefit my business or organization? [page 71]
5. Are there available assets? [page 71]

Technical Features and Configuration

6. What are the functionalities of each of the APIs? [page 71]
7. Which providers are supported by SAP HANA spatial services? [page 72]
8. Can SAP HANA spatial services be used for real-time geospatial data processing? [page 72]
9. Is real-time traffic considered with SAP HANA spatial services routing API? [page 72]
10. Can SAP HANA spatial services be used in a cloud environment or on-premises? [page 73]
11. Can the APIs be extended to offer more provider-specific functionality? [page 73]
12. If I have SAP HANA spatial services configured already, how do I switch between different providers? [page 73]
13. How do I need to configure my SAP Transportation Management (TM) system to use its built-in support for SAP HANA spatial services? [page 73]
14. Are there any limitations with using SAP HANA spatial service that I should be aware of? [page 73]
15. What support options are available if I encounter any issues? [page 74]
16. Does SAP HANA spatial services have a graphical user interface? [page 74]
17. How is the "various" parameter used? [page 74]

License and Pricing

18. How much does SAP HANA spatial services cost [page 74]
19. What pricing or licensing models are available for SAP HANA spatial services? [page 74]

[20. What is CPEA vs. PAYG? \[page 74\]](#)

[21. What does BYOL vs. OEM mean? \[page 75\]](#)

[22. How are transactions for SAP HANA spatial services aggregated into blocks? \[page 75\]](#)

[23. Are there any industry-specific use cases or success stories related to SAP HANA spatial services? \[page 75\]](#)

Integration and Availability

[24. What other SAP applications benefit from SAP HANA spatial services through standard integrations? \[page 75\]](#)

[25. Can SAP HANA spatial services only be used through other SAP products? \[page 75\]](#)

[26. Do I need a license with a third-party provider to consume SAP HANA spatial services? \[page 76\]](#)

[27. What are the requirements to consume SAP HANA spatial services? \[page 76\]](#)

[28. What is a service plan, and which one should I use? \[page 76\]](#)

[29. In which regions is SAP HANA spatial services available? \[page 76\]](#)

General Information

1. What is SAP HANA spatial services and what does it offer?

SAP HANA spatial services encompass a range of services operating within the SAP Business Technology Platform (SAP BTP). These services are designed to offer streamlined access via unified Application Programming Interfaces (APIs) to geocoding, mapping, and routing functionalities. For more information on the APIs, see question [6. What are the functionalities of each of the APIs? \[page 71\]](#). These capabilities are seamlessly integrated from third-party service providers. SAP HANA spatial services can be leveraged within an SAP application or a custom application.

2. How does SAP HANA spatial services enhance spatial data processing and analysis?

SAP HANA spatial services empower enterprises to harness geolocation data by seamlessly incorporating location-based services from third-party providers into their applications. This introduces the opportunity for organizations to integrate location-based data, enabling them to make well-informed strategic business choices based on valuable geographical insights. With this integration, they can be less worried about the complexities of integration, leveraging third-party service providers' geolocation services.

3. What are the key features of SAP HANA spatial services?

Key features of SAP HANA spatial services include:

1. Provides unified access layer and maintained APIs access to scalable third-party spatial services.
2. Adds a location-aware business solution aspect that can be readily used in SAP and non-SAP applications.
3. Provides a flexible choice of contracted geo location service providers with an ability to switch between them by only changing a single parameter value.
4. Caters to customer's flexibility to choose the usage licensing models better suited for them. For more information, see question [21. What does BYOL vs. OEM mean? \[page 75\]](#).

5. Enables simpler integration of geo location services in SAP or custom non-SAP applications because of the pre-existing integration/configuration in the background with the expert 3rd party service providers. Thus, also adding cost benefit on the customer's end.
6. Enables transparent transaction-based billing by supporting the CPEA/PAYG payment models of SAP BTP platform. For more information, see question [20. What is CPEA vs. PAYG? \[page 74\]](#).

4. How can SAP HANA spatial services benefit my business or organization?

The utilization of SAP HANA spatial services can offer substantial benefits to your business or organization by leveraging location-based data. This data provides a contextual framework for a wide array of business activities, ranging from understanding customer purchasing patterns to optimizing critical logistics operations.

The incorporation of SAP HANA spatial services empowers your business users with real-time spatial intelligence at every touchpoint of your operational processes. This capability enables effective decision-making in various scenarios, whether it's dispatching field personnel promptly, optimizing freight costs, enhancing customer service, or identifying the most efficient distribution routes. Overall, the service enhances your ability to make data-driven decisions by offering valuable insights derived from location-based data. The incorporation of SAP HANA spatial services into an existing business application includes custom app integration, and it is also pre-integrated into certain SAP applications.

5. Are there available assets?

1. Demos
 - [Access SAP HANA spatial services APIs using Visual Studio Code](#)
 - [Use SAP HANA spatial services Mapping API in a SAP Fiori application](#)
 - [SAP HANA spatial services in the context of SAP Transportation Management](#) 📄
2. Blogs
 - [How SAP HANA Spatial Services is used within SAP applications – Part 1 \(SAP Transportation Management\)](#) 📄
 - [How SAP HANA Spatial Services is used within SAP applications – Part 2 \(SAP Integrated Business Planning for Supply Chain\)](#) 📄
3. Tutorial
 - [Access SAP HANA Spatial Service APIs](#) 📄

Technical Features and Configuration

6. What are the functionalities of each of the APIs?

Here is a brief overview of each of the APIs, for more details please refer to our online help on SAP Help Portal: [SAP HANA Spatial Services](#).

1. **Mapping API** - Mapping API provides map tiles for building a base map to visualize georeferenced data. The Mapping APIs have two service variants:
 - **Raster Mapping Services** - used to visualize raster map tiles as a base map layer in a GIS system or a custom web application.
 - **Vector Mapping Services** - used to visualize vector map tiles with predefined styles as a base map layer in a GIS system or a custom Web application. The service exposes endpoints to fetch styles, single vector tiles, glyphs, and sprites.
Supported by providers – Esri.

2. **Routing API** - The Routing API provides services for geographical vehicle routing between source and destination locations from different providers. Customers will be able to select their provider of choice from a list of supported providers, without the need to change the request's structure when switching providers. Response structures are also independent of the chosen provider.

The Routing API has three endpoint variants:

- *POST /routing/v1/matrix* - Allows you to get information about routes between large sets of source and destination coordinates. Results include distances and travel durations calculated for all possible combinations of given sources and destinations.
- *POST /routing/v1/route* - Allows you to get detailed information about a calculated route between a source, a target and optional waypoint locations. This information includes distance, duration, and route shape.
- *GET /routing/v1/health* - Allows you to determine the service's current health status. Includes the option to determine a provider's health status, too.

3. **Geocoding API** - The Geocoding API provides REST services to transform multiple addresses at once into geographical positions (a pair of longitude and latitude coordinates) using different providers. Customers will be able to choose their service from a list of providers while request and response structures remain unchanged. The addresses can be provided as plain text, and for each address the best-matching result is returned in the form of a GeoJSON feature.

The Geocoding API provides two endpoint variants:

- *POST /geocoding/v1/geocode* - This endpoint converts a collection of addresses into geographical positions represented in the form of GeoJSON. For each supplied address, it generates a GeoJSON feature that includes the most accurate position, its relevance, and relevant address details. Even in cases of ambiguous addresses, the endpoint will still provide address information. If a given address cannot be geocoded successfully, an empty GeoJSON feature will be returned.
- *GET /geocoding/v1/health* - Allows you to determine the service's current health status. Includes the option to determine a provider's health status, too.

7. Which providers are supported by SAP HANA spatial services?

As of 2023/Q4, SAP HANA spatial services supports the following providers with the respective licensing models (for information, see question 21. [What does BYOL vs. OEM mean?](#) [page 75]):

1. [HERE](#) 📍 - BYOL, OEM
2. [TomTom](#) 📍 - BYOL (OEM: [Public Roadmap](#) 📍)
3. [Esri](#) 📍 - BYOL
4. [Trimble](#) 📍 - BYOL
5. [PTV](#) 📍 - BYOL ([Public Roadmap](#) 📍)

For an up to date list of supported providers and their licensing models, please explore the API descriptions available on the [SAP Help Portal](#).

8. Can SAP HANA spatial services be used for real-time geospatial data processing?

Yes, SAP HANA spatial services can be used for real-time data processing, and this depends on the provider's capabilities and the intended use case. All currently supported API requests are synchronous requests and return immediately with a response, which can be used immediately in your application. By default, SAP HANA spatial services accesses the most up-to-date data from its providers. In other words, each response is a snapshot of the data at the time of the request.

9. Is real-time traffic considered with SAP HANA spatial services routing API?

Yes, it is possible to request the calculation of a single route or of a matrix of routes taking real time traffic information into account. This is done via a specific parameter called "traffic" as part of the [routing request](#) 📍

that indicates whether the user wishes to have real time traffic information considered during the calculation of the route(s).

10. Can SAP HANA spatial services be used in a cloud environment or on-premises?

SAP HANA spatial services can be leveraged in both a cloud and on-premises environment. SAP HANA spatial services is a pure API (application programming interface) based service offering. Therefore, its services can be accessed from and leveraged in both a cloud and on-premises environment as long as there is access to SAP Business Technology Platform (BTP). Please note that SAP HANA spatial services is a cloud service and cannot be hosted or installed on-premise.

11. Can the APIs be extended to offer more provider-specific functionality?

Yes, this is possible. The APIs offered by SAP HANA spatial services cover the essential, most commonly used parameters between all the supported providers. All providers, though, support more than just this essential set of parameters to allow for more expertly crafted requests to cover the specific needs of as many different use cases as possible. This is especially true for Routing APIs. For that reason, SAP HANA spatial services offers a dedicated parameter called "various". It can be used to indicate provider-specific parameters in a routing request which are not part of the essential set of parameters offer by default by SAP HANA spatial services. Please check the [documentation](#) for a detailed description.

12. If I have SAP HANA spatial services configured already, how do I switch between different providers?

Switching between different providers is achieved by changing one parameter's value. In each of the APIs offered by SAP HANA spatial services, there is a request parameter called "provider". Enter the name of one of the supported providers you would like to switch to. Please note that you will need to adjust any provider-specific value provided as part of the "various" parameter (for more information, see question 11. [Can the APIs be extended to offer more provider-specific functionality? \[page 73\]](#)). Please also note that you will need to adjust the value of the "api_key" parameter in case you are using the BYOL licensing model. This is not required in case you are using the OEM licensing model (for more information, see question 21. [What does BYOL vs. OEM mean? \[page 75\]](#)). In that case, SAP HANA spatial services will take care to choose the correct credentials internally.

13. How do I need to configure my SAP Transportation Management (TM) system to use its built-in support for SAP HANA spatial services?

To configure the integration of SAP HANA spatial services within your TM system, please refer to the following SAP Notes for prerequisites, setup, and improvement instructions:

- [3058573 – Integration Note: SAP HANA Spatial Services / SAP Transportation Management – SAP ONE Support Launchpad](#) 
- [2751622 - Setup Guide for HSS in TM SCE Applications](#) 
Please refer to the end of "SAP Note 2751622 – Setup Guide for HSS in TM SCE Applications" for a comprehensive list of further SAP TM related SAP Notes for improvement and correction instructions.

14. Are there any limitations with using SAP HANA spatial service that I should be aware of?

- **Caching:** When using the Routing API relying on the established contract agreements between SAP and Here, caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed. Please be aware that this does not generally apply when bringing your own license (BYOL model). In this case, the terms of your own license apply. Please check the details of the service agreement available at the [SAP BTP Service Description Guide](#) .

- Using the OEM model (for information, see question 21. [What does BYOL vs. OEM mean? \[page 75\]](#)), it is forbidden to use the provided services to offer mobile routing functionality for on-board devices.
- Please be advised that the display of geographical data should adhere to local laws. This may include the requirement to display copyright information of the map provider.

15. What support options are available if I encounter any issues?

Customers can raise their issue tickets here at the following link: <https://me.sap.com/servicessupport>.

The Ticket Component is *BC-CP-CF-HSS*.

16. Does SAP HANA spatial services have a graphical user interface?

No. SAP HANA spatial services are a set of web APIs, there is no graphical user interface.

17. How is the "various" parameter used?

For information how to use the "various" parameter, see [Using the various Parameter \[page 60\]](#).

License and Pricing

18. How much does SAP HANA spatial services cost?

The cost of SAP HANA spatial services depends on the consumption model of choice (for information about CPEA or Pay-As-You-Go, see question 20. [What is CPEA vs. PAYG? \[page 74\]](#)), licensing model of choice (for information about OEM or BYOL, see question 21. [What does BYOL vs. OEM mean? \[page 75\]](#)), and the number of requests per API made by a customer.

The price is calculated by first considering the number of API requests made and then converting these into transactions. For detailed information on the conversion from request to transaction, please refer to [API Transaction Costs](#). The conversion used to convert from requests to transactions will depend on whether an OEM or BYOL license is leveraged. Once the total number of transactions are determined, you will fall within a tier under either the CPEA model or Pay-As-You-Go consumption model. For detailed information on the tier pricing, please refer to [Pricing in the Discovery Center](#).

For example: Consider a customer who has requested 250,000 addresses to be geocoded through the OEM licensing option. Assume 200 addresses to be geocoded = 1 transaction with an OEM contract. From there, we can identify what 250,000 geocoded requests correspond to in transactions by doing the following calculation: $250,000 \times (1/200) = 1,250$. Next, let us assume the monthly price in EUR for transactions in blocks of 1000 under the CPEA consumption model is EUR 130.00, and this price is valid for tier 1, or total transactions less than 10,000. Since 1,250 is less than 10,000, we fall under tier 1, and we have two blocks of 1,000. Thus, the monthly cost will be $130.00 \times 2 = 260.00$ EUR.

19. What pricing or licensing models are available for SAP HANA spatial services?

SAP HANA spatial services is available through either a Cloud Platform Enterprise Agreement (CPEA) or pay-as-you-go (PAYG) model. Additionally, customers have the option to leverage SAP's OEM license with one of our third-party providers or bring their own license (BYOL) with one of the designated providers.

20. What is CPEA vs. PAYG?

CPEA stands for Cloud Platform Enterprise Agreement and is a consumption-based commercial model for SAP Business Technology Platform (SAP BTP). The objective of the model is to create one rich and simple customer

experience for finding, trying, buying, and consuming cloud services from SAP and partners for net-new and installed based customers. It eliminates the need to license an SAP BTP service individually, and instead allows customers to buy credits that they can then put towards different BTP services. Charges incur on actual usage and will be charged monthly against the prior purchased cloud credit. For more details, please refer to this [CPEA FAQ](#).

On the other hand, Pay-As-You-Go (PAYG) is the alternative commercial option that lets you get started quickly with SAP BTP services and tools, paying only for those you need when you need them. No upfront budget outlay, financial commitment, or minimum usage is required, significantly reducing barriers to entry. For more information, please visit [What Is the Consumption-Based Commercial Model?](#).

The monthly cost under either the CPEA or PAYG models can be found on [Pricing](#).

21. What does BYOL vs. OEM mean?

BYOL and OEM represent two options customers have when it comes to choosing their licensing model.

With the OEM option, customers may leverage SAP's existing OEM license with providers. This is done by providing the term "SAP-KEY" as a value for the "api_key" parameter. With this option, the cost for requests made to SAP HANA spatial services include both the cost of using SAP's license and using SAP HANA spatial services.

With the BYOL (bring your own license) option, customers may leverage their own existing licenses with one of the designated providers. With this option, the costs for requests to SAP HANA spatial services cover only the costs of using SAP HANA spatial services. To use this option, the license customers bring must be with one of SAP HANA spatial services supported providers. For an up-to-date list of our supported providers, please explore the [SAP Help Portal](#).

22. How are transactions for SAP HANA spatial services aggregated into blocks?

Transactions for the usage of SAP HANA spatial services are counted and billed in blocks of 1000 (for more information, see question 18. [How much does SAP HANA spatial services cost \[page 74\]](#)), and transactions are aggregated across the entire Global Account before blocks are counted. For more information about Global Account's usage analytics and overviews, see question [Monitoring Usage and Consumption Costs in Your Global Account](#).

23. Are there any industry-specific use cases or success stories related to SAP HANA spatial services?

SAP HANA spatial services can be used in conjunction with SAP Transportation Management (TM) through a standard integration. All of the APIs offered by HSS are leveraged by TM to provide valuable functionalities such as visualizing routes on a map and using geo data to determine optimal routes.

Integration and Availability

24. What other SAP applications benefit from SAP HANA spatial services through standard integrations?

SAP Transportation Management as well as SAP Integrated Business Planning are two SAP applications that have standard integrations with SAP HANA spatial services.

25. Can SAP HANA spatial services only be used through other SAP products?

No, SAP HANA spatial services can be used through non-SAP products, too. SAP HANA spatial services is a pure web API-based service offering. As such, it can be used in any way that fits your needs without having

to go through any other SAP products. Its services are pre-integrated in several SAP applications, and it can generally be used in any business application and business context where you have a need for basemaps, routing, and/or geocoding functionality.

26. Do I need a license with a third-party provider to consume SAP HANA spatial services?

No, depending on the chosen provider such a license is not required. For more information, see questions [21. What does BYOL vs. OEM mean? \[page 75\]](#) and [7. Which providers are supported by SAP HANA spatial services? \[page 72\]](#).

27. What are the requirements to consume SAP HANA spatial services?

SAP HANA spatial services can be used with enterprise accounts based on the consumption-based model in two flavors:

- Cloud Platform Enterprise Agreement (CPEA)
- Pay-As-You-Go for SAP BTP

For more information, see [Basic Platform Concepts](#) and [What Is the Consumption-Based Commercial Model?](#).

For more information about onboarding SAP HANA spatial services, see [SAP HANA Spatial Services Onboarding Guide](#).

28. What is a service plan, and which one should I use?

To learn more about Service Plans, please refer to this explanation of the general concept of service plans for SAP Business Technology Platform ([Entitlements and Quotas](#)).

For SAP HANA spatial services, the only supported service plan is “lite”.

29. In which regions is SAP HANA spatial services available?

As of 2023/Q4, SAP HANA spatial services is available in the following regions:

- AWS: Australia (Sydney), Europe (Frankfurt), US East (VA)

Please refer to [Pricing in the Discovery Center](#)  to find an up-to-date list of regions where SAP HANA spatial services is available.

6 Accessibility Features in SAP HANA Spatial Services

To optimize your experience of SAP HANA spatial services, SAP Business Technology Platform provides features and settings that help you use the software efficiently.

Note

SAP HANA spatial services run on the SAP BTP cockpit. For this reason, the accessibility features for SAP BTP cockpit apply. For more information, see the accessibility documentation for SAP BTP cockpit on SAP Help Portal in [Usability and Accessibility Features in SAP BTP Cockpit](#).

For more information on screen reader support and keyboard shortcuts, see [Accessibility for End Users](#).

7 REFCON: HSS

Note

Caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Please check the details of the service agreement available at [PROFESSIONAL SERVICES AGREEMENT](#) .

Note

For further details on transaction billing and metering, see the API descriptions on SAP BTP Marketplace as well as the SAP HANA Spatial Services service description available at [PROFESSIONAL SERVICES AGREEMENT](#) .

Search here for new document versions: <https://www.sap.com/about/agreements/policies/cloud-platform.html> .

Note

Caching or storing any location data for the purpose of building a repository of location assets or scaling one request to serve multiple end users is prohibited. However, storing or caching for no more than 30 days only to the extent necessary for enabling or improving an end user's use of SAP HANA spatial services is allowed.

Please check the details of the service agreement available at [PROFESSIONAL SERVICES AGREEMENT](#) .

Note

For further details on transaction billing and metering, see the API descriptions on SAP BTP Marketplace as well as the SAP HANA Spatial Services service description available at [PROFESSIONAL SERVICES AGREEMENT](#) .

Caution

Starting March 1st, 2023, the two parameters **app_id** and **app_code** for provider **Here** will not be supported by SAP HANA spatial services anymore. Use the parameter **api_key**, instead. The reason for this change is that **Here** Technologies deprecated the method of authentication via **app_id** and **app_code**. For more information, see SAP Note [3202728](#) .

Caution

Starting October 24th, 2023, the two parameters **app_id** and **app_code** for provider **Here** are not supported by SAP HANA spatial services anymore. Use the parameter **api_key**, instead. The reason for this change is that **Here** Technologies deprecated the method of authentication via **app_id** and **app_code**. For more information, see SAP Note [3202728](#) .

The **credentials** object must include a **provider** and an **api_key**. Instead of an **api_key**, the combination of **app_id** and **app_code** is also possible, but is deprecated and will not be supported anymore after March 1st, 2023.

The **credentials** object must include a **provider** and an **api_key**.

This type of credentials will not be supported anymore after March 1st, 2023. Use the **api_key** parameter instead.

⚠ Caution

Starting October 24th, 2023, the two parameters **app_id** and **app_code** for provider **Here** are not supported by SAP HANA spatial services anymore. Use the parameter **api_key**, instead. The reason for this change is that **Here** Technologies deprecated the method of authentication via **app_id** and **app_code**. For more information, see SAP Note [3202728](#) 📄

The **credentials** object must include a **provider** and an **api_key**.

This type of credentials will not be supported anymore after March 1st, 2023. Use the **api_key** parameter instead.

SAP BTP cockpit SAP BTP cockpit

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, esri, ptv, tomtom, trimble

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, esri, ptv, tomtom, trimble

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, tomtom, trimble

Output values:

- **providers**: optional (string)
List of providers that are requested to health check; if value is set to "false", no providers are checked.
providers: here, tomtom, trimble

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2025 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.

