
HERE Location Services for Python

HERE Europe B.V.

Sep 29, 2021

GETTING STARTED

1	Prerequisites	3
2	Installation	5
3	Geocode	7
4	Reverse Geocode	9
5	Discover	11
6	Browse	13
7	Lookup	15
8	Autosuggest	17
9	Isoline Routing	19
10	Routing	21
11	Matrix Routing	29
12	Destination Weather	39
13	Tour Planning	43
14	here_location_services package	47
15	CHANGELOG	103
16	Contributing to HERE Location Services for Python	105
17	Indices and tables	107
	Python Module Index	109
	Index	111

A Python client for [HERE Location Services](#).

PREREQUISITES

Before you can install *HERE Location Services for Python*, run its test-suite, or use the example notebooks to make sure you meet the following prerequisites:

- A Python installation, 3.6+ recommended, with the *pip* command available to install dependencies.
- In order to use Location services APIs, authentication is required.

There are two ways to authenticate: * Authentication using an API key:

- For API key-based authentication you will need a HERE developer account, freely available under [HERE Developer Portal](#).
- An [API key](#) from the [HERE Developer Portal](#), in an environment variable named *LS_API_KEY* which you can set like this (with a valid value, of course):

```
$ export LS_API_KEY="MY-LS-API-KEY"
```

- OAuth token-based authentication:
 - * For OAuth token authentication you will need an account on the HERE Platform. To get more details on the HERE Platform account please check our documentation [Get a HERE account](#).

Once you have the account follow the below steps to get credentials:

- * Go to [HERE Platform Applications and Keys](#) and register a new app.
- * Create a key for the app and download the generated `credentials.properties` file.

The HERE platform generated app credentials should look similar to the example below:

```
here.user.id = <example_here>
here.client.id = <example_here>
here.access.key.id = <example_here>
here.access.key.secret = <example_here>
here.token.endpoint.url = <example_here>
```

You can provide your credentials using any of the following methods:

- * Default credentials

Place the credentials file into

For Linux/macOS: *\$HOME/.here/credentials.properties*

For Windows: *%USERPROFILE%.herecredentials.properties* Code snippet to instantiate LS object:

```
from here_location_services import LS

# platform credentials will be picked from the default credentials file's location mentioned above
# and api_key should not be set in env variable LS_API_KEY.
ls = LS()
```

* Environment Variables

You can override default credentials by assigning values to the following environment variables:

```
HERE_USER_ID
HERE_CLIENT_ID
HERE_ACCESS_KEY_ID
HERE_ACCESS_KEY_SECRET
HERE_TOKEN_ENDPOINT_URL
```

Code snippet to instantiate LS object:

```
from here_location_services import LS
from here_location_services import PlatformCredentials

ls = LS(platform_credentials=PlatformCredentials.from_env())
```

* Credentials File

You can specify any credentials file as an alternative to that found in `~/.here/credentials.properties`. An error is generated if there is no file present at the path, or if the file is not properly formatted. Code snippet to instantiate LS object:

```
from here_location_services import LS
from here_location_services import PlatformCredentials

platform_credentials = PlatformCredentials.from_credentials_file("<Path_to_file>")
ls = LS(platform_credentials=platform_credentials)
```


INSTALLATION

- Install HERE Location Services for Python with conda from the Anaconda [conda-forge channel](#) using the below command:

```
$ conda install -c conda-forge here-location-services
```

- Install HERE Location Services for Python from [PyPI](#):

```
$ pip install here-location-services
```

- Install HERE Location Services for Python from GitHub:

```
$ pip install -e git+https://github.com/heremaps/here-location-services-python  
↪#egg=here-location-services
```


GEOCODE

Geocode endpoint of [HERE Geocoding & Search API](#) is used to find the geo-coordinates of a known address, place, locality or administrative area, even if the query is incomplete or partly incorrect. It also returns a complete postal address string and address details.

3.1 Example

```
import os

from here_location_services import LS
from here_map_widget import Map, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

address = "Invalidenstr 116, 10115 Berlin, Germany"
gc_response = ls.geocode(query=address)

data = gc_response.to_geojson()
geo_layer = GeoJSON(data=data)

m = Map(api_key=LS_API_KEY, center=[52.53086, 13.38469], zoom=12)
m.add_layer(geo_layer)
m
```

3.2 Attributes

Attribute	Type	Doc
query	string	An input address query.
limit	int	Maximum number of results to be returned. Default value is 20.
lang	string	Language to be used for result rendering from a list of BCP47 compliant Language Codes.

REVERSE GEOCODE

To find the nearest address to specific geocoordinates, you can use [Reverse Geocode API](#).

4.1 Example

```
import os

from here_location_services import LS

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

rev_gc_response = ls.reverse_geocode(lat=52.53086, lng=13.38469)
rev_gc_response.items[0]["address"]["label"]
```

```
'Invalidenstraße 116, 10115 Berlin, Germany'
```

4.2 Attributes

Attribute	Type	Doc
lat	float	Latitude of point.
lng	float	Longitude of point.
limit	int	Maximum number of results to be returned. Default value is 1.
lang	string	Language to be used for result rendering from a list of BCP47 compliant Language Codes.

DISCOVER

Discover endpoint of [HERE Geocoding & Search API](#) simplifies searching for places. The user submits a free-form text request that returns candidate items (places and addresses related) in the order of intent matching relevance.

5.1 Example

```
import os

from here_location_services import LS
from here_map_widget import Map, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

disc_response = ls.discover(query="coffee", center=[52.53086, 13.38469], radius=1000)

data = disc_response.to_geojson()
geo_layer = GeoJSON(data=data, show_bubble=True, point_style={"radius": 6})

m = Map(api_key=LS_API_KEY, center=[52.53086, 13.38469], zoom=15)
m.add_layer(geo_layer)
m
```

5.2 Attributes

Attribute	Type	Doc
query	string	free-text query to search places.
center	list	A list of latitude and longitude representing the center for search query. Default value is <code>None</code> .
radius	int	A radius in meters along with center for searching places. Default value is <code>None</code> .
country_codes	list	A list of ISO 3166-1 alpha-3 country codes. Default value is <code>None</code> .
bounding_box	list	A bounding box, provided as west longitude, south latitude, east longitude, north latitude. Default value is <code>None</code> .
limit	int	Maximum number of results to be returned. Default value is <code>None</code> .
lang	string	Language to be used for result rendering from a list of BCP47 compliant Language Codes. Default value is <code>None</code> .

BROWSE

Browse endpoint of HERE Geocoding & Search API provides a structured search by filtering items by category and name at a given geo-position and radius. Items returned are places, streets or localities, ranked by increasing distance.

6.1 Example

```
import os

from here_location_services import LS
from here_location_services.config.search_config import PLACES_CATEGORIES
from here_map_widget import Map, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

browse_response = ls.browse(
    center=[52.53086, 13.38469],
    bounding_box=[13.08836, 52.33812, 13.761, 52.6755],
    categories=[PLACES_CATEGORIES.restaurant],
)

data = browse_response.to_geojson()
geo_layer = GeoJSON(data=data, show_bubble=True, point_style={"radius": 6})

m = Map(api_key=LS_API_KEY, center=[52.53086, 13.38469], zoom=15)
m.add_layer(geo_layer)
m
```

6.2 Attributes

Attribute	Type	Doc
center	list	A list of latitude and longitude representing the center for search query.
radius	int	A radius in meters along with center for searching places. Default value is <code>None</code> .
country_codes	list	A list of ISO 3166-1 alpha-3 country codes. Default value is <code>None</code> .
bounding_box	list	A bounding box, provided as west longitude, south latitude, east longitude, north latitude. Default value is <code>None</code> .
categories	list	A list of strings of category ids, defined as config <i>PLACES_CATEGORIES</i>
limit	int	Maximum number of results to be returned. Default value is <code>None</code> .
name	string	Full-text filter on POI names/titles. Default value is <code>None</code> .
lang	string	Language to be used for result rendering from a list of BCP47 compliant Language Codes. Default value is <code>None</code> .

LOOKUP

Every place or location object known by HERE has a location identifier or “ID”. Lookup endpoint of [HERE Geocoding & Search API](#) is used to look up a place by its HERE ID.

7.1 Example

```
import os

from here_location_services import LS

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

lookup = ls.lookup(location_id="here:pds:place:276u0vhj-b0bace6448ae4b0fbc1d5e323998a7d2
↪")

lookup.response
```

```
{'title': 'Flughafen Frankfurt-Hahn',
 'id': 'here:pds:place:276u0vhj-b0bace6448ae4b0fbc1d5e323998a7d2',
 'resultType': 'place',
 'address': {'label': 'Flughafen Frankfurt-Hahn, 55483 Lautzenhausen, Deutschland',
 'countryCode': 'DEU',
 'countryName': 'Deutschland',
 'stateCode': 'RP',
 'state': 'Rheinland-Pfalz',
 'countyCode': 'SIM',
 'county': 'Rhein-Hunsrück-Kreis',
 'city': 'Lautzenhausen',
 'postalCode': '55483'},
 'position': {'lat': 49.94802, 'lng': 7.27153},
 'access': [{'lat': 49.94571, 'lng': 7.26985}],
 'categories': [{'id': '400-4000-4581', 'name': 'Flughafen', 'primary': True}],
 'references': [{'supplier': {'id': 'core'}, 'id': '50481441'},
 {'supplier': {'id': 'venues'}, 'id': '1560904'}],
 'contacts': [{'phone': [{'value': '+496543509200'}],
 'www': [{'value': 'http://www.hahn-airport.de'}],
 'email': [{'value': 'info@hahn-airport.de'}]}]}
```

7.2 Attributes

Attribute	Type	Doc
location_id	string	A HERE location ID to lookup.
lang	string	Language to be used for result rendering from a list of BCP47 compliant Language Codes. Default value is <code>None</code> .

AUTOSUGGEST

Autosuggest endpoint of [HERE Geocoding & Search API](#) improves the user's search experience by allowing submittal of free-form, incomplete and misspelled addresses or place names to the endpoint.

8.1 Example

```
import os

from here_location_services import LS
from here_map_widget import Map, MarkerCluster, ObjectLayer

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

autosuggest_response = ls.autosuggest(
    query="bar",
    limit=5,
    at=["-13.163068,-72.545128"],
    terms_limit=3,
)
results = []
for item in autosuggest_response.items:
    if item["resultType"] == "place":
        results.append(
            dict(
                lat=item["position"]["lat"],
                lng=item["position"]["lng"],
                data=item["title"],
            )
        )
provider = MarkerCluster(data_points=results, show_bubble=True)
layer = ObjectLayer(provider=provider)
m = Map(
    api_key=LS_API_KEY,
    center=[-13.16, -72.52],
    zoom=14,
)
m.add_layer(layer)
m
```

8.2 Attributes

At-tribute	Type	Doc
query	string	A string for free-text query. Example: <i>res, rest</i>
at	list	optional Specify the center of the search context expressed as list of coordinates. One of <i>at</i> , <i>search_in_circle</i> or <i>search_in_bbox</i> is required. Parameters “at”, “search_in_circle” and “search_in_bbox” are mutually exclusive. Only one of them is allowed.
search_in_circle	optional	Search within a circular geographic area provided as latitude, longitude, and radius (in meters)
search_in_bbox	optional	Search within a rectangular bounding box geographic area provided as tuple of west longitude, south latitude, east longitude, north latitude
in_countries	list	optional Search within a specific or multiple countries provided as comma-separated ISO 3166-1 alpha-3 country codes. The country codes are to be provided in all uppercase. Must be accompanied by exactly one of <i>at</i> , <i>search_in_circle</i> or <i>search_in_bbox</i> .
limit	int	optional An integer specifying maximum number of results to be returned.
terms_limit	int	optional An integer specifying maximum number of Query Terms Suggestions to be returned.
lang	list	optional List of strings to select the language to be used for result rendering from a list of BCP 47 compliant language codes.
political_view	string	optional Toggle the political view by passing a string from POLITICAL_VIEW .
show	list	optional Select additional fields from SHOW . to be rendered in the response.

ISOLINE ROUTING

Isoline Routing API is used to calculate the area that a driver can reach within a given time or distance.

9.1 Example

```
import os
from datetime import datetime

from here_location_services import LS
from here_map_widget import Map, Marker, GeoJSON
from here_location_services.config.isoline_routing_config import RANGE_TYPE, ISOLINE_
↳ROUTING_TRANSPORT_MODE

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

iso_response = ls.calculate_isoline(
    origin=[52.53086, 13.38469],
    range="1800",
    departure_time=datetime.now(),
    range_type=RANGE_TYPE.time,
    transport_mode=ISOLINE_ROUTING_TRANSPORT_MODE.car,
)

data = iso_response.to_geojson()
geo_layer = GeoJSON(data=data)

iso_marker = Marker(lat=52.53086, lng=13.38469)

m = Map(api_key=LS_API_KEY, center=[52.53086, 13.38469], zoom=9)
m.add_layer(geo_layer)
m.add_object(iso_marker)
m
```

9.2 Attributes

Attribute	Type	Doc
range	string	A string representing a range of isoline, unit is defined by parameter <code>range_type</code> .
range_type	string	A string representing a type of <code>range</code> . Possible values are defined in RANGE_TYPE .
transport_mode	string	Represents transport mode to be used for the calculation of isolines. Values are defined in ISOLINE_ROUTING_TRANSPORT_MODE .
origin	list	optional A list of <code>latitude</code> and <code>longitude</code> for centers of the isoline request. The Isoline(s) will cover the region which can be reached from this point within given range. Cannot be used in combination with <code>destination</code> parameter.
departure_time	<code>datetime.datetime()</code>	optional departure time.
destination	list	optional A list of <code>latitude</code> and <code>longitude</code> for centers of the isoline request. The Isoline(s) will cover the region within the specified range that can reach this point. It cannot be used in combination with <code>origin</code> parameter.
arrival_time	<code>datetime.datetime()</code>	optional arrival time.
routing_mode	string	optional routing mode is defined in ROUTING_MODE
shape_max_points	integer	optional An integer to Limit the number of points in the resulting isoline geometry. If the isoline consists of multiple components, the sum of points from all components is considered. This parameter doesn't affect performance.
optimised_for	string	optional A string to specify how isoline calculation is optimized. Specify values from config: OPTIMISED_FOR
avoid_features	list	optional specify values from config: ISOLINE_ROUTING_AVOID_FEATURES to avoid features during isoline calculation.
truck	object of Truck	optional used to define truck options when transport mode is truck.
origin_place_options	PlaceOptions	optional place options for origin.
origin_waypoint_options	WayPointOptions	optional waypoint options for origin.
destination_place_options	PlaceOptions	optional place options for destination.
destination_waypoint_options	WayPointOptions	optional waypoint options for destination.

ROUTING

Routing API is used to calculate route between two or more locations.

10.1 Car route Example

Calculate car route between origin and destination.

```
import os

from here_location_services import LS
from here_location_services.config.routing_config import ROUTING_RETURN
from here_map_widget import Map, Marker, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

result = ls.car_route(
    origin=[52.51375, 13.42462],
    destination=[52.52332, 13.42800],
    return_results=[
        ROUTING_RETURN.polyline,
        ROUTING_RETURN.elevation,
        ROUTING_RETURN.instructions,
        ROUTING_RETURN.actions,
    ],
)
geo_json = result.to_geojson()
data = geo_json
geo_layer = GeoJSON(data=data, style={"lineWidth": 5})

m = Map(api_key=LS_API_KEY, center=[52.5207, 13.4283], zoom=14)
origin_marker = Marker(lat=52.51375, lng=13.42462)
dest_marker = Marker(lat=52.52332, lng=13.42800)
m.add_layer(geo_layer)
m.add_object(origin_marker)
m.add_object(dest_marker)
m
```

10.1.1 Attributes

Attribute	Type	Doc
origin	list	A list of latitude and longitude of origin point of route.
destination	list	A list of latitude and longitude of destination point of route.
via	list	A list of here_location_services.config.routing_config.Via objects.
origin_place_options	PlaceOptions	optional place options for origin.
destination_place_options	PlaceOptions	optional place options for destination.
destination_waypoint_options	WaypointOptions	optional way point options for destination.
departure_time	<code>datetime.datetime()</code>	optional departure time.
routing_mode	string	optional routing mode is defined in ROUTING_MODE
alternatives	int	optional number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
units	string	optional representing units of measurement used in guidance instructions, valid values are <code>metric</code> and <code>imperial</code> , default value is <code>metric</code> .
lang	string	optional preferred language of the response. The value should comply with the IETF BCP 47, default is <code>en-US</code> .
return_results	list	optional list of strings, values are defined in ROUTING_RETURN
spans	list	optional list of strings, values are defined in ROUTING_SPANS

10.2 Bicycle route Example

Calculate bicycle route between origin and destination.

```
import os

from here_location_services import LS
from here_location_services.config.routing_config import ROUTING_RETURN
from here_map_widget import Map, Marker, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

result = ls.bicycle_route(
    origin=[52.51375, 13.42462],
    destination=[52.52332, 13.42800],
    return_results=[
        ROUTING_RETURN.polyline,
        ROUTING_RETURN.elevation,
        ROUTING_RETURN.instructions,
        ROUTING_RETURN.actions,
    ],
)
geo_json = result.to_geojson()
data = geo_json
```

(continues on next page)

(continued from previous page)

```

geo_layer = GeoJSON(data=data, style={"lineWidth": 5})

m = Map(api_key=LS_API_KEY, center=[52.5207, 13.4283], zoom=14)
origin_marker = Marker(lat=52.51375, lng=13.42462)
dest_marker = Marker(lat=52.52332, lng=13.42800)
m.add_layer(geo_layer)
m.add_object(origin_marker)
m.add_object(dest_marker)
m

```

10.2.1 Attributes

Attribute	Type	Doc
origin	list	A list of latitude and longitude of origin point of route.
destination	list	A list of latitude and longitude of destination point of route.
via	list	A list of here_location_services.config.routing_config.Via objects.
origin_place_options	PlaceOptions	optinal place options for origin.
destination_place_options	PlaceOptions	optional place options for destination.
destination_waypoint_options	WayPointOptions	optional way point options for destination.
departure_time	datetime.datetime()	optional departure time.
routing_mode	string	optional routing mode is defined in ROUTING_MODE
alternatives	int	optional number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
units	string	optional representing units of measurement used in guidance instructions, valid values are <code>metric</code> and <code>imperial</code> , default value is <code>metric</code> .
lang	string	optional preferred language of the response. The value should comply with the IETF BCP 47, default is <code>en-US</code> .
return_results	list	optional list of strings, values are defined in ROUTING_RETURN
spans	list	optional list of strings, values are defined in ROUTING_SPANS

10.3 Pedestrian route Example

Calculate pedestrian route between origin and destination.

```

import os

from here_location_services import LS
from here_location_services.config.routing_config import ROUTING_RETURN
from here_map_widget import Map, Marker, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

```

(continues on next page)

(continued from previous page)

```

result = ls.pedestrian_route(
    origin=[52.51375, 13.42462],
    destination=[52.52332, 13.42800],
    return_results=[
        ROUTING_RETURN.polyline,
        ROUTING_RETURN.elevation,
        ROUTING_RETURN.instructions,
        ROUTING_RETURN.actions,
    ],
)
geo_json = result.to_geojson()
data = geo_json
geo_layer = GeoJSON(data=data, style={"lineWidth": 5})

m = Map(api_key=LS_API_KEY, center=[52.5207, 13.4283], zoom=14)
origin_marker = Marker(lat=52.51375, lng=13.42462)
dest_marker = Marker(lat=52.52332, lng=13.42800)
m.add_layer(geo_layer)
m.add_object(origin_marker)
m.add_object(dest_marker)
m

```

10.3.1 Attributes

Attribute	Type	Doc
origin	list	A list of latitude and longitude of origin point of route.
destination	list	A list of latitude and longitude of destination point of route.
via	list	A list of here_location_services.config.routing_config.Via objects.
origin_place_options	PlaceOptions	optinal place options for origin.
destination_place_options	PlaceOptions	optional place options for destination.
destination_waypoint_options	WayPointOptions	optional way point options for destination.
departure_time	datetime.datetime()	optional departure time.
routing_mode	string	optional routing mode is defined in <code>ROUTING_MODE</code>
alternatives	int	optional number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
units	string	optional representing units of measurement used in guidance instructions, valid values are <code>metric</code> and <code>imperial</code> , default value is <code>metric</code> .
lang	string	optional preferred language of the response. The value should comply with the IETF BCP 47, default is <code>en-US</code> .
return_results	list	optional list of strings, values are defined in ROUTING_RETURN
spans	list	optional list of strings, values are defined in ROUTING_SPANS

10.4 Scooter route Example

Calculate the scooter route between origin and destination.

```
import os

from here_location_services import LS
from here_location_services.config.routing_config import ROUTING_RETURN
from here_map_widget import Map, Marker, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

result = ls.scooter_route(
    origin=[52.51375, 13.42462],
    destination=[52.52332, 13.42800],
    return_results=[
        ROUTING_RETURN.polyline,
        ROUTING_RETURN.elevation,
        ROUTING_RETURN.instructions,
        ROUTING_RETURN.actions,
    ],
)
geo_json = result.to_geojson()
data = geo_json
geo_layer = GeoJSON(data=data, style={"lineWidth": 5})

m = Map(api_key=LS_API_KEY, center=[52.5207, 13.4283], zoom=14)
origin_marker = Marker(lat=52.51375, lng=13.42462)
dest_marker = Marker(lat=52.52332, lng=13.42800)
m.add_layer(geo_layer)
m.add_object(origin_marker)
m.add_object(dest_marker)
m
```

10.4.1 Attributes

Attribute	Type	Doc
origin	list	A list of latitude and longitude of origin point of route.
destination	list	A list of latitude and longitude of destination point of route.
via	list	A list of here_location_services.config.routing_config.Via objects.
origin_place_options	PlaceOptions	optional place options for origin.
destination_place_options	PlaceOptions	optional place options for destination.
destination_waypoint_options	WayPointOptions	optional way point options for destination.
scooter	Scooter	optional additional attributes for scooter route.
departure_time	datetime.datetime()	optional departure time.
routing_mode	string	optional routing mode is defined in ROUTING_MODE
alternatives	int	optional number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
units	string	optional representing units of measurement used in guidance instructions, valid values are metric and imperial , default value is metric .
lang	string	optional preferred language of the response. The value should comply with the IETF BCP 47, default is en-US .
return_results	list	optional list of strings, values are defined in ROUTING_RETURN
spans	list	optional list of strings, values are defined in ROUTING_SPANS

10.5 Truck route Example

Calculate truck route between origin and destination.

```
import os

from here_location_services import LS
from here_location_services.config.routing_config import ROUTING_RETURN
from here_map_widget import Map, Marker, GeoJSON

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

result = ls.truck_route(
    origin=[52.51375, 13.42462],
    destination=[52.52332, 13.42800],
    return_results=[
        ROUTING_RETURN.polyline,
        ROUTING_RETURN.elevation,
        ROUTING_RETURN.instructions,
        ROUTING_RETURN.actions,
    ],
)
geo_json = result.to_geojson()
```

(continues on next page)

(continued from previous page)

```

data = geo_json
geo_layer = GeoJSON(data=data, style={"lineWidth": 5})

m = Map(api_key=LS_API_KEY, center=[52.5207, 13.4283], zoom=14)
origin_marker = Marker(lat=52.51375, lng=13.42462)
dest_marker = Marker(lat=52.52332, lng=13.42800)
m.add_layer(geo_layer)
m.add_object(origin_marker)
m.add_object(dest_marker)
m

```

10.5.1 Attributes

Attribute	Type	Doc
origin	list	A list of latitude and longitude of origin point of route.
destination	list	A list of latitude and longitude of destination point of route.
via	list	A list of here_location_services.config.routing_config.Via objects.
origin_place_options	PlaceOptions	optimal place options for origin.
destination_place_options	PlaceOptions	optional place options for destination.
destination_waypoint_options	WayPointOptions	optional way point options for destination.
departure_time	datetime.datetime()	optional departure time.
routing_mode	string	optional routing mode is defined in <code>ROUTING_MODE</code>
alternatives	int	optional number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
units	string	optional representing units of measurement used in guidance instructions, valid values are <code>metric</code> and <code>imperial</code> , default value is <code>metric</code> .
lang	string	optional preferred language of the response. The value should comply with the IETF BCP 47, default is <code>en-US</code> .
return_results	list	optional list of strings, values are defined in ROUTING_RETURN
spans	list	optional list of strings, values are defined in ROUTING_SPANS

MATRIX ROUTING

Matrix Routing API v8 is used to calculate routing matrices of up to 10,000 origins and 10,000 destinations. A routing matrix is a matrix with rows labeled by origins and columns by destinations. Each entry of the matrix is travel time or distance from the origin to the destination.

The Matrix Routing service provides the following features:

- A large number of origins and destinations (up to 10,000)
- Live traffic and historical speed patterns
- Car, Truck, Pedestrian and Bicycle modes
- Truck attributes such as dimensions, weight, tunnel restrictions, and more
- Avoiding areas and routing features, e.g., toll roads, ferries, and motorways.
- The choice between synchronous and asynchronous APIs for flexible result downloads

Calculation of routing matrices in one of the following modes:

- Flexible
- Region
- Profile

The values of the `region_definition` and `profile` parameters determine which mode is used. The following table describes the capabilities and limitations of each mode.

Table 1: Modes

Mode	region_definition parameter	profile parameter provided?	Custom Options & Time Awareness (incl. live traffic)	Unlimited region
Flexible	world	no	yes	yes
Region	one of: Circle, boundingBox, polygon, autoCircle	no	yes	no, origins and destinations must be within a region of max 400 km diameter
Profile	world	yes	no	yes

Note that the combination of specifying a profile along with a `region_definition` not equal to world is not allowed.

11.1 Flexible Mode

Flexible mode provides capabilities such as Custom options, Time Awareness (including Live Traffic), Unlimited Region but it has a limited Matrix Size. Given a list of origins and a list of destinations, the service computes the shortest travel times or distances between every pair of origin and destination. These results make up the entries of the [routing matrix](#).

In order to provide support for custom routing options, time awareness, and routes of arbitrary length, Flexible Mode cannot benefit from the optimizations that give Region and Profile modes their high performance. Due to this performance limitation, Flexible Mode requests are limited to:

- at most 15 origins and 100 destinations (15 x 100)
- at most 100 origins and 1 destination (100 x 1)

11.1.1 Formulating a request

The flexible mode is utilized when:

the region definition = WorldRegion and no profile parameter is specified The service applies live and historical traffic information unless explicitly disabled by setting departureTime to the special value any.

Below is an example of a 3x3 matrix request with the following origins and destinations:

- San Francisco at (37.76, -122.42)
- New York at (40.63, -74.09)
- Austin at (30.26, -97.74)

11.1.2 Example

```
import os

from here_location_services import LS
from here_location_services.config.matrix_routing_config import (
    WorldRegion,
    MATRIX_ATTRIBUTES,
)

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

origins = [
    {"lat": 37.76, "lng": -122.42},
    {"lat": 40.63, "lng": -74.09},
    {"lat": 30.26, "lng": -97.74},
]
region_definition = WorldRegion()
matrix_attributes = [MATRIX_ATTRIBUTES.distances, MATRIX_ATTRIBUTES.travelTimes]

result = ls.matrix(
    origins=origins,
    region_definition=region_definition,
```

(continues on next page)

(continued from previous page)

```
    matrix_attributes=matrix_attributes,  
)  
result.matrix
```

```
{'numOrigins': 3,  
 'numDestinations': 3,  
 'travelTimes': [0, 154467, 92781, 154876, 0, 93882, 94302, 93955, 0],  
 'distances': [0, 4701596, 2831834, 4701970, 0, 2829527, 2834742, 2803612, 0]}
```

11.1.3 Attributes

Table 2: Attributes

Attribute	Type	Doc
origins	list	A list of dictionaries containing lat and long for origin points.
region_definition	object	use one of the: <i>CircleRegion</i> <i>BoundingBoxRegion</i> <i>PolygonRegion</i> <i>AutoCircleRegion</i> <i>WorldRegion</i>
async_req	bool	If set to True requests will be sent to asynchronous matrix routing API else It will be sent to synchronous matrix routing API. For larger matrices, or longer routes, or routes in denser road networks, it is recommended to set to True.
destinations	list	A list of dictionaries containing lat and long for destination points. When no destinations are specified the matrix is assumed to be quadratic with origins used as destinations.
profile	string	Use values from config: <i>PROFILE</i>
departure_time	datetime.datetime object with timezone	When it is not specified, it is implicitly assumed to be the current time. The special value any enforces non time-aware routing.
routing_mode	string	Use values from config: <i>ROUTING_MODE</i>
transport_mode	string	Use values from config: <i>ROUTING_TRANSPORT_MODE</i>
avoid_features	list	Use values from config: <i>AVOID_FEATURES</i>
avoid_areas	list	Use object of <i>AvoidBoundingBox</i> to define avoid areas.
truck	object of <i>Truck</i>	Used to define truck options when transport mode is truck
matrix_attributes	list	Use values from config: <i>MATRIX_ATTRIBUTES</i>

11.2 Region Mode

This section refers to calculating matrices with custom options using a limited-sized region. The region is limited to the max. 400km diameter. By restricting the calculation to a specific region of at most 400 km diameter, it is possible to specify different options to take into account during calculation. The service applies live and historical traffic information unless explicitly disabled by setting *departure_time* to the special value *any*.

Region Mode supports:

- Custom options
- Time Awareness (including Live Traffic), using a snapshot of time at departure
- Matrix Sizes up to 10,000 x 10,000
- Region limited to max. 400km diameter

11.2.1 BoundingBox region definition

Below is an example of a simple 3 x 3 matrix in Berlin, Germany with these origins and destinations:

- Alexanderplatz at (52.52103, 13.41268)
- Brandenburg Gate at (52.51628, 13.37771)
- Tempelhof Field at (52.47342, 13.40357)

To calculate a car distance matrix, you can use the below code. Since the request does not specify a destinations list, the origins are taken as destinations and the resulting matrix is a 3 x 3 matrix. The region definition is a bounding box around the points with a small margin added to be able to properly route in the vicinity of the points. By default, the service calculates a travel times matrix, but since we want to get distances in the response instead of times, the request specifies the *matrix_attributes* property with the value *distances*.

11.2.2 Example

```
import os
from here_location_services import LS
from here_location_services.config.matrix_routing_config import (
    BoundingBoxRegion,
    AutoCircleRegion,
    MATRIX_ATTRIBUTES,
    WorldRegion,
)

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY) # Create Location Services object using API KEY.

origins = [
    {"lat": 52.52103, "lng": 13.41268},
    {"lat": 52.51628, "lng": 13.37771},
    {"lat": 52.47342, "lng": 13.40357},
]
region_definition = BoundingBoxRegion(north=52.53, south=52.46, west=13.35, east=13.42)
matrix_attributes = [MATRIX_ATTRIBUTES.distances]
```

(continues on next page)

(continued from previous page)

```

result = ls.matrix(
    origins=origins,
    region_definition=region_definition,
    matrix_attributes=matrix_attributes,
    async_req=True,
)
result.matrix

```

```

{'numOrigins': 3,
 'numDestinations': 3,
 'distances': [0, 2924, 7259, 3526, 0, 8328, 7546, 7628, 0]}

```

11.2.3 AutoCircle region definition

Instead of defining a bounding box around the origins, you can request for a circle to be automatically derived. The request below is for the same as the one above, but using the AutoCircle feature. Since the margin field is not provided, the service uses a default value of 10 kilometers.

11.2.4 Example

```

import os
from here_location_services import LS
from here_location_services.config.matrix_routing_config import (
    BoundingBoxRegion,
    AutoCircleRegion,
    MATRIX_ATTRIBUTES,
    WorldRegion,
)

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY) # Create Location Services object using API KEY.

origins = [
    {"lat": 52.52103, "lng": 13.41268},
    {"lat": 52.51628, "lng": 13.37771},
    {"lat": 52.47342, "lng": 13.40357},
]
region_definition = AutoCircleRegion()
matrix_attributes = [MATRIX_ATTRIBUTES.distances]

result = ls.matrix(
    origins=origins,
    region_definition=region_definition,
    matrix_attributes=matrix_attributes,
    async_req=True
)

result.response

```

```
{'matrixId': 'acd6bfa4-d952-49e8-9b11-30f9c36da53b',
 'matrix': {'numOrigins': 3,
            'numDestinations': 3,
            'distances': [0, 2924, 7259, 3526, 0, 8328, 7546, 7628, 0]},
 'regionDefinition': {'type': 'circle',
                      'center': {'lat': 52.497225, 'lng': 13.395195},
                      'radius': 12900}}
```

11.2.5 Attributes

Table 3: Attributes

Attribute	Type	Doc
origins	list	A list of dictionaries containing lat and long for origin points.
region_definition	object	use one of the: CircleRegion BoundingBoxRegion PolygonRegion AutoCircleRegion WorldRegion
async_req	bool	If set to True requests will be sent to asynchronous matrix routing API else It will be sent to synchronous matrix routing API. For larger matrices, or longer routes, or routes in denser road networks, it is recommended to set to True.
destinations	list	A list of dictionaries containing lat and long for destination points. When no destinations are specified the matrix is assumed to be quadratic with origins used as destinations.
profile	string	Use values from config: PROFILE
departure_time	<code>datetime.datetime</code> object with timezone	When it is not specified, it is implicitly assumed to be the current time. The special value <code>any</code> enforces non time-aware routing.
routing_mode	string	Use values from config: ROUTING_MODE
transport_mode	string	Use values from config: ROUTING_TRANSPORT_MODE
avoid_features	list	Use values from config: AVOID_FEATURES
avoid_areas	list	Use object of AvoidBoundingBox to define avoid areas.
truck	object of <code>Truck</code>	Used to define truck options when transport mode is truck
matrix_attributes	list	Use values from config: MATRIX_ATTRIBUTES

11.3 Profile Mode

Profile mode supports:

- unlimited region
- Matrix Sizes up to 10,000 x 10,000

Profile mode does not support:

- Custom Options
- Time Awareness (including Live Traffic)

This section refers to calculating matrices with routes of arbitrary length, using one of the supported profiles. If you want to define custom options, see [Flexible Mode](#).

The special variant `world` needs to be set as region definition. No additional request options or `departure_time` can be provided except for `matrix_attributes`. Below is an example of a 7 x 7 matrix request with these origins and destinations:

- Berlin at (52.54, 13.40)
- Kyiv at (50.43, 30.52)
- London at (51.50, -0.08)
- Madrid at (40.40, -3.68)
- Moscow at (55.75, 37.60)
- Paris at (48.87, 2.33)
- Rome at (41.90, 12.48)

To calculate a car distance matrix, you can use the code below. Since the request does not specify the destinations array, the origins are taken as destinations and the resulting matrix is a 7 x 7 matrix. The region definition is the special variant `world`. In the request, we use the profile `carFast` which uses transport mode `car` and optimizes the route calculations for travel time. By default, the service calculates a travel times matrix, but since we want to get distances in the response instead of times, the request specifies the `matrix_attributes` property with the value `distances`.

11.3.1 Example

```
import os
from here_location_services import LS
from here_location_services.config.matrix_routing_config import (
    BoundingBoxRegion,
    AutoCircleRegion,
    MATRIX_ATTRIBUTES,
    PROFILE,
    WorldRegion
)

LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY) # Create Location Services object using API KEY.

origins = [
    {"lat": 52.54, "lng": 13.40},
    {"lat": 50.43, "lng": 30.52},
    {"lat": 51.50, "lng": -0.08},
```

(continues on next page)

(continued from previous page)

```
    {"lat": 40.40, "lng": -3.68},  
    {"lat": 55.75, "lng": 37.60},  
    {"lat": 48.87, "lng": 2.33},  
    {"lat": 41.90, "lng": 12.48},  
]  
  
profile = PROFILE.carFast  
region_definition = WorldRegion()  
matrix_attributes = [MATRIX_ATTRIBUTES.distances]  
  
result = ls.matrix(  
    origins=origins,  
    region_definition=region_definition,  
    matrix_attributes=matrix_attributes,  
    async_req=True  
)  
result.to_distances_matrix()
```

11.3.2 Attributes

Table 4: Attributes

Attribute	Type	Doc
origins	list	A list of dictionaries containing lat and long for origin points.
region_definition	object	use one of the: <i>CircleRegion</i> <i>BoundingBoxRegion</i> <i>PolygonRegion</i> <i>AutoCircleRegion</i> <i>WorldRegion</i>
async_req	bool	If set to True requests will be sent to asynchronous matrix routing API else It will be sent to synchronous matrix routing API. For larger matrices, or longer routes, or routes in denser road networks, it is recommended to set to True.
destinations	list	A list of dictionaries containing lat and long for destination points. When no destinations are specified the matrix is assumed to be quadratic with origins used as destinations.
profile	string	Use values from config: <i>PROFILE</i>
departure_time	datetime.datetime object with timezone	When it is not specified, it is implicitly assumed to be the current time. The special value any enforces non time-aware routing.
routing_mode	string	Use values from config: <i>ROUTING_MODE</i>
transport_mode	string	Use values from config: <i>ROUTING_TRANSPORT_MODE</i>
avoid_features	list	Use values from config: <i>AVOID_FEATURES</i>
avoid_areas	list	Use object of <i>AvoidBoundingBox</i> to define avoid areas.
truck	object of Truck	Used to define truck options when transport mode is truck
matrix_attributes	list	Use values from config: <i>MATRIX_ATTRIBUTES</i>

DESTINATION WEATHER

Destination Weather API provides weather forecasts and reports on current weather conditions. It also provides information on severe weather alerts along a specified route or a single car location.

12.1 Example

```
import os
from here_location_services import LS
from here_map_widget import Map, MarkerCluster, ObjectLayer
from here_location_services.config.dest_weather_config import DEST_WEATHER_PRODUCT

LS_API_KEY = os.environ.get("LS_API_KEY")
ls = LS(api_key=LS_API_KEY)

result1 = ls.get_dest_weather(
    at=[19.1503, 72.8530],
    products=[DEST_WEATHER_PRODUCT.observation]
)

results = []
m = Map(
    api_key=LS_API_KEY,
    center=[19.1621, 73.0008],
    zoom=7,
)
for observation in result1.places[0]["observations"]:
    results.append(
        dict(
            lat=observation["place"]["location"]["lat"],
            lng=observation["place"]["location"]["lng"],
            data=observation["description"] + " " + str(observation["temperature"]),
            ↪+ "C",
        )
    )

provider = MarkerCluster(data_points=results, show_bubble=True)
layer = ObjectLayer(provider=provider)
m.add_layer(layer)
m
```

12.2 Attributes

At-tribute	Type	Doc
prod-ucts	list of <i>DestWeatherProduct</i>	List of strings identifying the type of report to obtain.
at	list	optional A list of latitude and longitude specifying the area covered by the weather report.
query	str	optional Free text query. Examples: “125, Berliner, berlin”, “Beacon, Boston”
zipcode	str	optional ZIP code of the location. This parameter is supported only for locations in the United States of America.
hourly_data	datetime.datetime()	optional Date for which hourly forecasts are to be retrieved.
one_observation	Boolean	optional Boolean, if set to true, the response only includes the closest location. Only available when the <i>product</i> parameter is set to <i>DEST_WEATHER_PRODUCT.observation</i> .
lan-guage	str	optional Defines the language used in the descriptions in the response.
units	<i>DestWeatherUnits</i>	optional Defines whether units or imperial units are used in the response.

12.3 Getting Weather Alerts

Can be used to get information on severe weather alerts along a specified route or a single car location.

```
import os
from here_location_services import LS
from geojson import Point
from datetime import datetime

LS_API_KEY = os.environ.get("LS_API_KEY")
ls = LS(api_key=LS_API_KEY)
result = ls.get_weather_alerts(
    geometry=Point(coordinates=[15.256, 23.456]),
    start_time=datetime.now(),
    width=3000,
)

print(result)
```

```
{"features": [], "type": "FeatureCollection"}
```

12.4 Attributes

At-tribute	Type	Doc
geome-try	Point or LineString	Point or LineString defining the route or a single location
start_time	datetime.datetime()	Start time of the event
id	str	optional Unique weather alert id.
weather_severity	WeatherSeverity	optional Defines the severity of the weather event
weather_type	WeatherType	optional Defines the type of the weather event
country	str	optional String for ISO-3166-1 2-letter country code.
end_time	datetime.datetime()	optional End time of the event. If not present, warning is valid until it is not removed from the feed by national weather institutes (valid until warning is present in the response)
width	int	optional int.

TOUR PLANNING

Tour Planning API The HERE Tour Planning API allows you to dynamically optimize routes for multiple vehicles visiting a set of locations given real-life constraints, such as limited capacity in a vehicle or delivery time windows.

The HERE Tour Planning API supports the following use cases:

- **Capacitated vehicle routing problem:** You can use Tour Planning API to take your vehicle capacity into account when routing your vehicles.
- **Vehicle routing problem with time windows:** With Tour Planning API, you can schedule your vehicles to visit depots only when they are available.
- **Multi-depot vehicle routing problem:** In a simple vehicle routing problem, all vehicles start from the same location. In the multi-depot vehicle routing problem, vehicles start from multiple depots and return to their depots of origin at the end of their assigned tours.
- **Open vehicle routing problem:** Do you work with drivers who have their vehicles and do not return to a planned location after their drop-offs? With Tour Planning API, you can schedule open vehicle routing where your drivers can return to their homes after work.
- **Heterogeneous or mixed fleet VRP:** Tour Planning API supports routing multiple types of vehicles with different gas mileage, cost of driving, capacity, and more. For example, your fleet can include passenger vehicles and even specialized trucks with fridges in one route.
- **Pick up and delivery vehicle routing problem:** With Tour Planning API, you can schedule a vehicle to pick up and deliver an item in one route.
- **Vehicle routing problem with priorities:** Do you have jobs that must be served, such as today, and others that could also be delayed until, such as until tomorrow, and ideally you would like to prevent those priority jobs from being skipped in cases where fleet capacity or shift durations do not allow to serve all jobs, but at the same time you would like to serve as many non-priority jobs as possible? With Tour Planning API, you can define jobs to be with high priority internally the algorithms will try to avoid skipping those high priority jobs and will skip low priority jobs first in scenarios where it is impossible to serve all jobs due to constraints. The priority of a job does not imply its order in the route, as in the position of a high priority job might be anywhere in the route and not necessarily before lower priority jobs.

13.1 Example

```
import os

from here_location_services import LS
from here_location_services.config.tour_planning_config import (
    VEHICLE_MODE,
    Fleet,
    Job,
    JobPlaces,
    Plan,
    Relation,
    VehicleProfile,
    VehicleType,
)
LS_API_KEY = os.environ.get("LS_API_KEY") # Get API KEY from environment.
ls = LS(api_key=LS_API_KEY)

fleet = Fleet(
    vehicle_types=[
        VehicleType(
            id="09c77738-1dba-42f1-b00e-eb63da7147d6",
            profile_name="normal_car",
            costs_fixed=22,
            costs_distance=0.0001,
            costs_time=0.0048,
            capacity=[100, 5],
            skills=["fridge"],
            amount=1,
            shift_start={
                "time": "2020-07-04T09:00:00Z",
                "location": {"lat": 52.5256, "lng": 13.4542},
            },
            limits={"maxDistance": 20000, "shiftTime": 21600},
            shift_end={
                "location": {"lat": 52.5256, "lng": 13.4542},
                "time": "2020-07-04T18:00:00Z",
            },
            shift_breaks=[
                {
                    "duration": 1800,
                    "times": ["2020-07-04T11:00:00Z", "2020-07-04T13:00:00Z"],
                }
            ],
        )
    ],
    vehicle_profiles=[VehicleProfile(name="normal_car", vehicle_mode=VEHICLE_MODE.car)],
)

plan = Plan(
    jobs=[
```

(continues on next page)

(continued from previous page)

```

Job(
    id="4bbc206d-1583-4266-bac9-d1580f412ac0",
    pickups=[
        JobPlaces(
            duration=180,
            demand=[10],
            location=(52.53088, 13.38471),
            times=[["2020-07-04T10:00:00Z", "2020-07-04T12:00:00Z"]],
        )
    ],
    deliveries=[
        JobPlaces(
            duration=300,
            demand=[10],
            location=(52.53088, 13.38471),
            times=[["2020-07-04T14:00:00Z", "2020-07-04T16:00:00Z"]],
        )
    ],
    skills=["fridge"],
    priority=2,
)
],
relations=[
    Relation(
        type="sequence",
        jobs=["departure", "4bbc206d-1583-4266-bac9-d1580f412ac0", "arrival"],
        vehicle_id="09c77738-1dba-42f1-b00e-eb63da7147d6_1",
    )
],
)

# Synchronous Solving
response = ls.solve_tour_planning(
    fleet=fleet, plan=plan, id="7f3423c2-784a-4983-b472-e14107d5a54a"
)
print(response)

# Asynchronous Solving
async_response = ls.solve_tour_planning(
    fleet=fleet,
    plan=plan,
    id="7f3423c2-784a-4983-b472-e14107d5a54a",
    is_async=True
)
print(async_response)

```

```

{"problemId": "7f3423c2-784a-4983-b472-e14107d5a54a", "statistic": {"cost": 68.394,
    ↳ "distance": 11782, "duration": 9420, "times": {"break": 1800, "driving": 1320, "serving":
    ↳ 480, "waiting": 5820}}, "tours": [{"statistic": {"cost": 68.394, "distance": 11782,
    ↳ "duration": 9420, "times": {"break": 1800, "driving": 1320, "serving": 480, "waiting":
    ↳ 5820}}, "stops": [{"activities": [{"jobId": "departure", "type": "departure"}], "load
    ↳ ": [0], "location": {"lat": 52.5256, "lng": 13.4542}, "time": {"arrival": "2020-07-
    ↳ 04T11:38:50Z", "departure": "2020-07-04T11:38:50Z"}}, {"activities": [{"jobId":
    ↳ "4bbc206d-1583-4266-bac9-d1580f412ac0", "location": {"lat": 52.53088, "lng": 13.38471},
    ↳ "time": {"end": "2020-07-04T11:53:00Z", "start": "2020-07-04T11:50:00Z", "type":
    ↳ "pickup"}, {"jobId": "break", "location": {"lat": 52.53088, "lng": 13.38471}, "time": {
    ↳ "end": "2020-07-04T12:23:00Z", "start": "2020-07-04T11:53:00Z", "type": "break"}, {
    ↳ "jobId": "4bbc206d-1583-4266-bac9-d1580f412ac0", "location": {"lat": 52.53088, "lng":
    ↳ 13.38471}, "time": {"end": "2020-07-04T14:05:00Z", "start": "2020-07-04T12:23:00Z"}].

```

(continued from previous page)

```
{
  "problemId": "7f3423c2-784a-4983-b472-e14107d5a54a",
  "statistic": {
    "cost": 68.394,
    "distance": 11782,
    "duration": 9420,
    "times": {
      "break": 1800,
      "driving": 1320,
      "serving": 480,
      "waiting": 5820
    }
  },
  "tours": [
    {
      "statistic": {
        "cost": 68.394,
        "distance": 11782,
        "duration": 9420,
        "times": {
          "break": 1800,
          "driving": 1320,
          "serving": 480,
          "waiting": 5820
        }
      },
      "stops": [
        {
          "activities": [
            {
              "jobId": "departure",
              "type": "departure"
            }
          ],
          "load": [0],
          "location": {
            "lat": 52.5256,
            "lng": 13.4542
          },
          "time": {
            "arrival": "2020-07-04T11:38:50Z",
            "departure": "2020-07-04T11:38:50Z"
          }
        },
        {
          "activities": [
            {
              "jobId": "4bbc206d-1583-4266-bac9-d1580f412ac0",
              "location": {
                "lat": 52.53088,
                "lng": 13.38471
              },
              "time": {
                "end": "2020-07-04T11:53:00Z",
                "start": "2020-07-04T11:50:00Z"
              },
              "type": "pickup"
            },
            {
              "jobId": "break",
              "location": {
                "lat": 52.53088,
                "lng": 13.38471
              },
              "time": {
                "end": "2020-07-04T12:23:00Z",
                "start": "2020-07-04T11:53:00Z"
              },
              "type": "break"
            },
            {
              "jobId": "4bbc206d-1583-4266-bac9-d1580f412ac0",
              "location": {
                "lat": 52.53088,
                "lng": 13.38471
              },
              "time": {
                "end": "2020-07-04T14:05:00Z",
                "start": "2020-07-04T12:23:00Z"
              },
              "type": "delivery"
            }
          ],
          "load": [0],
          "location": {
            "lat": 52.53088,
            "lng": 13.38471
          },
          "time": {
            "arrival": "2020-07-04T11:50:00Z",
            "departure": "2020-07-04T14:05:00Z"
          }
        },
        {
          "activities": [
            {
              "jobId": "arrival",
              "type": "arrival"
            }
          ],
          "load": [0],
          "location": {
            "lat": 52.5256,
            "lng": 13.4542
          },
          "time": {
            "arrival": "2020-07-04T14:15:50Z",
            "departure": "2020-07-04T14:15:50Z"
          }
        }
      ],
      "typeId": "09c77738-1dba-42f1-b00e-eb63da7147d6",
      "vehicleId": "09c77738-1dba-42f1-b00e-eb63da7147d6_1"
    }
  ]
}
```

13.2 Attributes

Attribute	Type	Doc
fleet	Fleet	A fleet represented by various vehicle types for serving jobs.
plan	Plan	Represents the list of jobs to be served.
id	string	optional A unique identifier of an entity. Avoid referencing any confidential or personal information as part of the Id.
optimization_traffic	string	optional "liveOrHistorical" "historicalOnly" "automatic" Specifies what kind of traffic information should be considered for routing
optimization_waiting_time	Dict	optional Configures departure time optimization which tries to adapt the starting time of the tour in order to reduce waiting time as a consequence of a vehicle arriving at a stop before the starting time of the time window defined for serving the job.
is_async	bool	optional Solves the problem Asynchronously

HERE_LOCATION_SERVICES PACKAGE

This module contains top level imports for `here_location_services` package.

14.1 Submodules

14.1.1 `here_location_services.__version__` module

Project version information.

14.1.2 `here_location_services.apis` module

This module contains base classes for accessing the Location Services RESTful APIs.

class `here_location_services.apis.Api`(*api_key=None, auth=None, proxies=None, country='row'*)

Bases: `object`

A base class for low-level HTTP RESTful API client for location services.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

__init__(*api_key=None, auth=None, proxies=None, country='row'*)

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

_get_url_string()

Get url string from config based on type of country.

For china url string ends with `hereapi.cn` and for rest of the countries denoted by `row` it is `hereapi.com`.

Raises Exception – If `api_key` not found in credentials.

Return type str

__add_api_key_in_params(*params*)

Add api_key in query params dictionary.

Returns Dict.

Parameters *params* (Dict) –

Return type Dict

get(*url*, *params=None*, ***kwargs*)

Send HTTP GET request.

Parameters

- **url** (str) – A string to represent URL.
- **params** (Optional[Dict]) – An optional dict for query params.
- **kwargs** – An optional extra arguments.

Returns requests.Response object.

post(*url*, *data*, *params=None*)

Send HTTP POST request.

Parameters

- **url** (str) – A string to represent URL.
- **data** (Dict) – A dictionary to represent the post data
- **params** (Optional[Dict]) – An optional dict for query params.

Returns requests.Response object.

14.1.3 here_location_services.config module

This module defines all the project level configs.

14.1.4 here_location_services.config.base_config module

This module defines all the base classes which will be used for configuration classes of various APIs.

class here_location_services.config.base_config.Bunch(***kwargs*)

Bases: dict

A class for dot notation implementation of dictionary.

__init__(***kwargs*)

class here_location_services.config.base_config.RoutingMode(***kwargs*)

Bases: [here_location_services.config.base_config.Bunch](#)

A Class to define constant values for Routing Modes.

fast: Route calculation from start to destination optimized by travel time. In many cases, the route returned by the fast mode may not be the route with the fastest possible travel time. For example, the routing service may favor a route that remains on a highway, even if a faster travel time can be achieved by taking a detour or shortcut through an inconvenient side road.

short: Route calculation from start to destination disregarding any speed information. In this mode, the distance of the route is minimized, while keeping the route sensible. This includes, for example, penalizing turns. Because of that, the resulting route will not necessarily be the one with minimal distance.

```
here_location_services.config.base_config.ROUTING_MODE = {'fast': 'fast', 'short': 'short'}
```

Use this config for routing_mode of routing API. Example: for fast routing_mode use ROUTING_MODE.fast.

```
class here_location_services.config.base_config.ShippedHazardousGoods(**kwargs)
```

Bases: [here_location_services.config.base_config.Bunch](#)

A class to define the constant values for truck option shippedHazardousGoods.

```
here_location_services.config.base_config.SHIPPED_HAZARDOUS_GOODS = {'combustible': 'combustible', 'corrosive': 'corrosive', 'explosive': 'explosive', 'flammable': 'flammable', 'gas': 'gas', 'harmfulToWater': 'harmfulToWater', 'organic': 'organic', 'other': 'other', 'poison': 'poison', 'poisonousInhalation': 'poisonousInhalation', 'radioactive': 'radioactive'}
```

Use this config for shipped_hazardous_goods attribute of Truck options of matrix Routing API. Example: for explosive shipped_hazardous_goods use SHIPPED_HAZARDOUS_GOODS.explosive.

```
class here_location_services.config.base_config.Truck(shipped_hazardous_goods=None,
                                                    gross_weight=None, weight_per_axle=None,
                                                    height=None, width=None, length=None,
                                                    tunnel_category=None, axle_count=None,
                                                    truck_type='straight', trailer_count=0)
```

Bases: object

A class to define different truck options which will be used during route calculation. Truck options should be used when transport_mode is truck.

Parameters

- **shipped_hazardous_goods** (Optional[List]) –
- **gross_weight** (Optional[int]) –
- **weight_per_axle** (Optional[int]) –
- **height** (Optional[int]) –
- **width** (Optional[int]) –
- **length** (Optional[int]) –
- **tunnel_category** (Optional[str]) –
- **axle_count** (Optional[int]) –
- **truck_type** (str) –
- **trailer_count** (int) –

```
__init__(shipped_hazardous_goods=None, gross_weight=None, weight_per_axle=None, height=None,
         width=None, length=None, tunnel_category=None, axle_count=None, truck_type='straight',
         trailer_count=0)
```

Object Initializer.

Parameters

- **shipped_hazardous_goods** (Optional[List]) – List of hazardous materials in the vehicle. valid values for hazardous materials can be used from config SHIPPED_HAZARDOUS_GOODS

- **gross_weight** (*Optional[int]*) – Total vehicle weight, including trailers and shipped goods, in kilograms. Should be greater than or equal to zero.
- **weight_per_axle** (*Optional[int]*) – Vehicle weight per axle, in kilograms. Should be greater than or equal to zero.
- **height** (*Optional[int]*) – Vehicle height, in centimeters. Should be in range [0, 5000]
- **width** (*Optional[int]*) – Vehicle width, in centimeters. Should be in range [0, 5000]
- **length** (*Optional[int]*) – Vehicle length, in centimeters. Should be in range [0, 5000]
- **tunnel_category** (*Optional[str]*) – A string for category of tunnel. Valid values are “B”, “C”, “D”, “E”. Specifies the [cargo tunnel restriction code](#). The route will pass only through tunnels of less restrictive categories.
- **axle_count** (*Optional[int]*) – Total number of axles that the vehicle has. Should be in the range [2, 255].
- **truck_type** (*str*) – A string to represent the type of truck.
- **trailer_count** (*int*) – Number of trailers attached to the vehicle.

```
class here_location_services.config.base_config.PlaceOptions(course=None,
                                                            sideof_street_hint=None,
                                                            match_sideof_street=None,
                                                            namehint=None, radius=None,
                                                            min_course_distance=None)
```

Bases: object

A class to define PlaceOptions for origin/ via/ destination.

Various options can be found here:

[PlaceOptions](#).

Parameters

- **course** (*Optional[int]*) –
- **sideof_street_hint** (*Optional[List[float]]*) –
- **match_sideof_street** (*Optional[str]*) –
- **namehint** (*Optional[str]*) –
- **radius** (*Optional[int]*) –
- **min_course_distance** (*Optional[int]*) –

```
__init__(course=None, sideof_street_hint=None, match_sideof_street=None, namehint=None,
         radius=None, min_course_distance=None)
```

Object Initializer.

Parameters

- **course** (*Optional[int]*) – An int representing degrees clock-wise from north. Indicating the desired direction at the place. E.g. 90 indicating east. This is defined in constant `ROUTE_COURSE`.
- **sideof_street_hint** (*Optional[List[float]]*) – A list of latitude and longitude. Indicating the side of the street that should be used.
- **match_sideof_street** (*Optional[str]*) – Specifies how the location set by `sideof_street_hint` should be handled. If this is set then `sideof_street_hint` should also be set. There are two valid values for `match_sideof_street`:

always: Always prefer the given side of street.

onlyIfDivided: Only prefer using side of street set by `sideof_street_hint` in case the street has dividers. This is the default behavior.

These values are maintained as config in: [ROUTE_MATCH_SIDE_OF_STREET](#)

- **namehint** (*Optional[str]*) – A string for the router to look for the place with the most similar name. This can e.g. include things like: North being used to differentiate between interstates I66 North and I66 South, Downtown Avenue being used to correctly select a residential street.
- **radius** (*Optional[int]*) – In meters Asks the router to consider all places within the given radius as potential candidates for route calculation. This can be either because it is not important which place is used, or because it is unknown. Radius more than 200 meters are not supported.
- **min_course_distance** (*Optional[int]*) – In meters Asks the routing service to try to find a route that avoids actions for the indicated distance. E.g. if the origin is determined by a moving vehicle, the user might not have time to react to early actions.

__repr__()

Return string representation of this instance.

```
class here_location_services.config.base_config.WayPointOptions(stop_duration=None,
                                                                pass_through=None)
```

Bases: object

A class to define PlaceOptions for via/ destination.

Various options can be found here:

[PlaceOptions](#).

Parameters

- **stop_duration** (*Optional[int]*) –
- **pass_through** (*Optional[bool]*) –

__init__(*stop_duration=None, pass_through=None*)

Parameters

- **stop_duration** (*Optional[int]*) –
- **pass_through** (*Optional[bool]*) –

__repr__()

Return string representation of this instance.

14.1.5 here_location_services.config.autosuggest_config module

This module defines all the configs which will be required as inputs to autosuggest API.

```
class here_location_services.config.autosuggest_config.SearchCircle(lat, lng, radius)
```

Bases: object

A class to define SearchCircle

Results will be returned if they are located within the specified circular area defined by its center and radius(in meters).

Parameters

- **lat** (*float*) –
- **lng** (*float*) –
- **radius** (*int*) –

__init__(*lat, lng, radius*)

Parameters

- **lat** (*float*) –
- **lng** (*float*) –
- **radius** (*int*) –

class here_location_services.config.autosuggest_config.**PoliticalView**(**kwargs)

Bases: [here_location_services.config.base_config.Bunch](#)

A Class to define constant values for political view

RUS: expressing the Russian view on Crimea

SRB: expressing the Serbian view on Kosovo, Vukovar and Sarengrad Islands

MAR: expressing the Moroccan view on Western Sahara

here_location_services.config.autosuggest_config.POLITICAL_VIEW = {'MAR': 'MAR', 'RUS': 'RUS', 'SRB': 'SRB'}

Use this config for political_view of Autosuggest API. Example: for RUS political_view use POLITICAL_VIEW. RUS.

class here_location_services.config.autosuggest_config.**Show**(**kwargs)

Bases: [here_location_services.config.base_config.Bunch](#)

A Class to define constant values for showing additional fields to be rendered in the response.

phonemes: Renders phonemes for address and place names into the results.

tz: BETA: Renders result items with additional time zone information. Please note that this may impact latency significantly.

here_location_services.config.autosuggest_config.SHOW = {'phonemes': 'phonemes', 'tz': 'tz'}

Use this config for show of Autosuggest API. Example: for RUS show use SHOW.phonemes.

14.1.6 here_location_services.config.isoline_routing_config module

This module defines all the configs which will be required as inputs to Isoline routing API.

class here_location_services.config.isoline_routing_config.**IsolineRoutingTransportMode**(**kwargs)

Bases: [here_location_services.config.base_config.Bunch](#)

A class to define constant attributes for mode of transport to be used for the calculation of the route.

- car
- truck
- pedestrian


```
here_location_services.config.isoline_routing_config.ISOLINE_ROUTING_TRANSPORT_MODE =
{'car': 'car', 'pedestrian': 'pedestrian', 'truck': 'truck'}
```

Use this config for transport_mode of isoline routing API. Example: for car transport_mode use ISOLINE_ROUTING_TRANSPORT_MODE.car.

```
class here_location_services.config.isoline_routing_config.RangeType(**kwargs)
```

Bases: [here_location_services.config.base_config.Bunch](#)

A Class to define constant values for specifying the type of range for Isoline Routing API

distance: Units in meters

time: Units in seconds

consumption: Units in Wh

```
here_location_services.config.isoline_routing_config.RANGE_TYPE = {'consumption':
'consumption', 'distance': 'distance', 'time': 'time'}
```

Use this config s optimised_for of isoline routing API. Example: for optimising for balanced mode use OPTIMISED_FOR.balanced.

```
class here_location_services.config.isoline_routing_config.OptimisedFor(**kwargs)
```

Bases: [here_location_services.config.base_config.Bunch](#)

A Class to define constant values for optimising calculation for Isoline Routing API

quality: Calculation of isoline focuses on quality, that is, the graph used for isoline calculation has higher granularity generating an isoline that is more precise.

performance: Calculation of isoline is performance-centric, quality of isoline is reduced to provide better performance.

balanced: Calculation of isoline takes a balanced approach averaging between quality and performance.

```
here_location_services.config.isoline_routing_config.OPTIMISED_FOR = {'balanced':
'balanced', 'performance': 'performance', 'quality': 'quality'}
```

Use this config s optimised_for of isoline routing API. Example: for optimising for balanced mode use OPTIMISED_FOR.balanced.

```
class here_location_services.config.isoline_routing_config.IsolineRoutingAvoidFeatures(**kwargs)
```

Bases: [here_location_services.config.base_config.Bunch](#)

A class to define values for features to avoid features during isoline calculation.

```
here_location_services.config.isoline_routing_config.ISOLINE_ROUTING_AVOID_FEATURES =
{'carShuttleTrain': 'carShuttleTrain', 'controlledAccessHighway':
'controlledAccessHighway', 'difficultTurns': 'difficultTurns', 'dirtRoad': 'dirtRoad',
'ferry': 'ferry', 'tollRoad': 'tollRoad', 'tunnel': 'tunnel'}
```

Use this config for avoid_features of isoline API. Example: for tollRoad avoid_features use ISOLINE_ROUTING_AVOID_FEATURES.tollRoad.

14.1.7 here_location_services.config.dest_weather_config module

This module defines all the configs which will be required as inputs to Destination Weather API.

```
class here_location_services.config.dest_weather_config.DestWeatherProduct(**kwargs)
```

Bases: [here_location_services.config.base_config.Bunch](#)

A class to define constant attributes for product parameter identifying the type of report to obtain.

observation: current weather conditions from the eight closest locations to the specified location

forecast_7days: morning, afternoon, evening and night weather forecasts for the next seven days.

forecast_7days_simple: daily weather forecasts for the next seven days

forecast_hourly: hourly weather forecasts for the next seven days

alerts: forecasted weather alerts for the next 24 hours

nws_alerts: all active watches and warnings for the US and Canada

```
here_location_services.config.dest_weather_config.DEST_WEATHER_PRODUCT = {'alerts':  
'alerts', 'forecast7days': 'forecast7days', 'forecast7daysSimple':  
'forecast7daysSimple', 'forecastHourly': 'forecastHourly', 'nwsAlerts': 'nwsAlerts',  
'observation': 'observation'}
```

Use this config for products` of Destination Weather API. Example: for forecastHourly product use DEST_WEATHER_PRODUCT.forecastHourly.

```
class here_location_services.config.dest_weather_config.DestWeatherUnits(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant attributes for units parameter identifying units of measurement used.

metric: Follow metric system of measurements. Default.

imperial: Follow imperial system of measurements

```
here_location_services.config.dest_weather_config.DEST_WEATHER_UNITS = {'imperial':  
'imperial', 'metric': 'metric'}
```

Use this config for units` of Destination Weather API. Example: for metric units use DEST_WEATHER_UNITS.metric.

```
class here_location_services.config.dest_weather_config.WeatherSeverity(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define the severity of the weather event

insignificant: Event doesn't have significance by nature

no_alerts: There are no alerts for the location

minor: Minor Severity, the event is potentially dangerous but not usual

medium: Medium Severity, the event is dangerous

high: High Severity, The event is very dangerous

emergency: Emergency. Take immediate action to protect life.

```
here_location_services.config.dest_weather_config.WEATHER_SEVERITY = {'emergency': 5,  
'high': 4, 'insignificant': 0, 'medium': 3, 'minor': 2, 'no_alerts': 1}
```

Use this config for weather_severity` of get weather alerts endpoint. Example: for high severity events use WEATHER_SEVERITY.high.

```
class here_location_services.config.dest_weather_config.WeatherType(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define the type of the weather event

```
here_location_services.config.dest_weather_config.WEATHER_TYPE = {'air_quality': 9,  
'avalanche': 11, 'civil_danger': 24, 'coastal_event': 23, 'dust_storm': 13,  
'earthquake': 14, 'evacuation': 25, 'extremely_high_temperature': 1,  
'extremely_low_temperature': 2, 'fire_danger': 15, 'flood': 16, 'fog': 3,  
'gigh_uv_index': 18, 'hazardous_material': 26, 'high_waves': 17, 'ice': 4,  
'ice_in_waterway': 22, 'low_water': 19, 'radiological_hazard': 27, 'rain': 5,  
'shelter_in_place': 28, 'smoke': 20, 'snow': 6, 'thunderstorm': 7, 'tsunami': 12,  
'volcanic_ashfall': 10, 'volcano': 21, 'warning': 29, 'wind': 8}
```

Use this config for weather_type` of get weather alerts endpoint of Destination Weather API. Example: for

fog weather type use `WEATHER_TYPE.fog`.

14.1.8 `here_location_services.config.routing_config` module

This module defines all the configs which will be required as inputs to routing APIs.

class `here_location_services.config.routing_config.RoutingReturn(**kwargs)`

Bases: `here_location_services.config.base_config.Bunch`

A class to define constant attributes which are included in routing API's response as part of the data representation of route or section.

- `polyline` - Polyline for the route in Flexible Polyline Encoding. Either a 2D polyline (without elevation specified) or a 3D polyline with the 3rd dimension type Elevation (with elevation specified).
- `actions` - Actions (such as maneuvers or tasks) that must be taken to complete the section.
- `instructions` - Include instructions in returned actions. Instructions are localized to the requested language.
- `summary` - Include a summary for the section.
- `travelSummary` - Include a summary for the travel portion of the section.
- `turnByTurnActions` - Include all information necessary to support turn by turn guidance to complete the section.
- `mlDuration` - Use a region-specific machine learning model to calculate route duration. Disclaimer: This parameter is currently in beta release, and is therefore subject to breaking changes.
- `elevation` - Include elevation information in coordinate and geometry types. See e.g. `polyline` or `location`.
- `routeHandle` - Encode calculated route and return a handle which can be used with `routes/{routeHandle}` to decode the route at a later point in time.
- `incidents` - Include a list of all incidents applicable to each section.

Following restrictions apply when specifying return parameter:

- If `actions` is requested, then `polyline` must also be requested as well.
- If `instructions` is requested, then `actions` must also be requested as well.
- If `turnByTurnActions` is requested, then `polyline` must also be requested as well.
- If at least one attribute is requested within the `spans` parameter, then `polyline` must be requested as well.

```
here_location_services.config.routing_config.ROUTING_RETURN = {'actions': 'actions',
'elevation': 'elevation', 'incidents': 'incidents', 'instructions': 'instructions',
'mlDuration': 'mlDuration', 'passthrough': 'passthrough', 'polyline': 'polyline',
'routeHandle': 'routeHandle', 'summary': 'summary', 'travelSummary': 'travelSummary',
'turnByTurnActions': 'turnByTurnActions'}
```

Use this config for return attributes from routing API for param `return_results`. Example: To return `polyline` in results use `ROUTING_RETURN.polyline`.

class `here_location_services.config.routing_config.RoutingSpans(**kwargs)`

Bases: `here_location_services.config.base_config.Bunch`

A class to define constant attributes which are included in the response spans.

`walkAttributes` `streetAttributes` `carAttributes` `truckAttributes` `scooterAttributes` `names`
`length` `duration` `baseDuration` `countryCode` `functionalClass` `routeNumbers` `speedLimit` `maxSpeed`
`dynamicSpeedInfo` `segmentId` `segmentRef` `consumption`.

```
here_location_services.config.routing_config.ROUTING_SPANS = {'baseDuration':  
'baseDuration', 'carAttributes': 'streetAttributes', 'consumption': 'consumption',  
'countryCode': 'countryCode', 'duration': 'duration', 'dynamicSpeedInfo':  
'dynamicSpeedInfo', 'functionalClass': 'functionalClass', 'length': 'length',  
'maxSpeed': 'maxSpeed', 'names': 'names', 'routeNumbers': 'routeNumbers',  
'scooterAttributes': 'scooterAttributes', 'segmentId': 'segmentId', 'segmentRef':  
'segmentRef', 'speedLimit': 'speedLimit', 'streetAttributes': 'streetAttributes',  
'truckAttributes': 'truckAttributes', 'walkAttributes': 'walkAttributes'}
```

Use this config for spans of routing API. Example: for walkAttributes routing_mode use ROUTING_SPANS.walkAttributes.

```
class here_location_services.config.routing_config.RoutingTransportMode(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant attributes for mode of transport to be used for the calculation of the route.

- car
- truck
- pedestrian
- bicycle
- scooter

```
here_location_services.config.routing_config.ROUTING_TRANSPORT_MODE = {'bicycle':  
'bicycle', 'car': 'car', 'pedestrian': 'pedestrian', 'scooter': 'scooter', 'truck':  
'truck'}
```

Use this config for transport_mode of routing API. Example: for car transport_mode use ROUTING_TRANSPORT_MODE.car.

```
class here_location_services.config.routing_config.RouteCourse(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant attributes for Course option.

```
class here_location_services.config.routing_config.RouteMatchSideOfStreet(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant attributes for matchSideOfStreet.

```
here_location_services.config.routing_config.ROUTE_MATCH_SIDE_OF_STREET = {'always':  
'always', 'onlyIfDivided': 'onlyIfDivided'}
```

Use this config for transport_mode of routing API. Example: for car transport_mode use ROUTING_TRANSPORT_MODE.car.

```
class here_location_services.config.routing_config.AvoidFeatures(**kwargs)  
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant values for features to avoid during route calculation.

```
here_location_services.config.routing_config.AVOID_FEATURES = {'carShuttleTrain':  
'carShuttleTrain', 'controlledAccessHighway': 'controlledAccessHighway',  
'difficultTurns': 'difficultTurns', 'dirtRoad': 'dirtRoad', 'ferry': 'ferry',  
'seasonalClosure': 'seasonalClosure', 'tollRoad': 'tollRoad', 'tunnel': 'tunnel'}
```

Use this config for match_sideof_street for PlaceOptions. Example: for match_sideof_street always use ROUTE_MATCH_SIDE_OF_STREET.always.

```
class here_location_services.config.routing_config.Scooter(allow_highway)  
    Bases: object
```

A class to define attributes specific for the scooter route.

Scooter specific parameters:

`allowHighway`: Specifies whether the scooter is allowed on the highway or not. This parameter is optional. If not provided, then by default scooter is not allowed to use highway. There is a similar parameter `avoid[features]=controlledAccessHighway` to disallow highway usage. `avoid[features]` takes precedence so if this parameter is also used then scooters are not allowed to use highways even if `allowHighway` is used with value as true. Possible values: true: scooter is allowed to use the highway. false: scooter is not allowed to use the highway.

Parameters `allow_highway (bool)` –

`__init__ (allow_highway)`

Parameters `allow_highway (bool)` –

`__repr__ ()`

Return string representation of this instance.

```
class here_location_services.config.routing_config.Via(lat, lng, place_options=None,
                                                       waypoint_options=None)
```

Bases: object

A class to define via waypoint.

A via waypoint is a location between origin and destination. The route will do a stop at the via waypoint.

Parameters

- `lat (float)` –
- `lng (float)` –
- `place_options` (Optional[`here_location_services.config.base_config.PlaceOptions`]) –
- `waypoint_options` (Optional[`here_location_services.config.base_config.WayPointOptions`]) –

`__init__ (lat, lng, place_options=None, waypoint_options=None)`

Parameters

- `lat (float)` –
- `lng (float)` –
- `place_options` (Optional[`here_location_services.config.base_config.PlaceOptions`]) –
- `waypoint_options` (Optional[`here_location_services.config.base_config.WayPointOptions`]) –

14.1.9 here_location_services.config.matrix_routing_config module

This module defines all the configs which will be required as inputs to matrix Routing API.

class here_location_services.config.matrix_routing_config.**CircleRegion**(*center, radius*)
Bases: object

A class to define attributes of Circle Region for Matrix Routing API.

Parameters

- **center** (*Dict*) –
- **radius** (*int*) –

__init__(*center, radius*)

Parameters

- **center** (*Dict*) –
- **radius** (*int*) –

__repr__()

Return string representation of this instance.

class here_location_services.config.matrix_routing_config.**BoundingBoxRegion**(*north, south, west, east*)

Bases: object

A class to define attributes of BoundingBox Region for Matrix Routing API.

Parameters

- **north** (*float*) –
- **south** (*float*) –
- **west** (*float*) –
- **east** (*float*) –

__init__(*north, south, west, east*)

Parameters

- **north** (*float*) –
- **south** (*float*) –
- **west** (*float*) –
- **east** (*float*) –

__repr__()

Return string representation of this instance.

class here_location_services.config.matrix_routing_config.**PolygonRegion**(*outer*)
Bases: object

A class to define attributes of Polygon Region for Matrix Routing API.

Parameters **outer** (*List*) –

__init__(*outer*)

Parameters *outer* (*List*) –

__repr__()

Return string representation of this instance.

class `here_location_services.config.matrix_routing_config.AutoCircleRegion(margin=10000)`
Bases: `object`

A class to define attributes of AutoCircle Region for Matrix Routing API.

Parameters *margin* (*int*) –

__init__(*margin=10000*)

Parameters *margin* (*int*) –

__repr__()

Return string representation of this instance.

class `here_location_services.config.matrix_routing_config.WorldRegion`
Bases: `object`

A class to define attributes of World Region for Matrix Routing API.

__init__()

class `here_location_services.config.matrix_routing_config.Profile(**kwargs)`
Bases: `here_location_services.config.base_config.Bunch`

A class to define constant values for Profile of Matrix Routing API.

A profile ID enables the calculation of matrices with routes of arbitrary length. All profiles explicitly set departureTime to any and require that the obligatory request parameter REGION_DEFINITION_TYPE is set to world.

Table 1: Profiles

Profile ID	Description
carFast	Car with fast routing mode
carShort	Car with short routing mode
truckFast	Truck with fast routing mode
pedestrian	Pedestrian transport mode
bicycle	Bicycle transport mode

`here_location_services.config.matrix_routing_config.PROFILE = {'bicycle': 'bicycle', 'carFast': 'carFast', 'carShort': 'carShort', 'pedestrian': 'pedestrian', 'truckFast': 'truckFast'}`

Use this config for profile of matrix Routing API. Example: for carFast profile use PROFILE.carFast.

class `here_location_services.config.matrix_routing_config.MatrixAttributes(**kwargs)`
Bases: `here_location_services.config.base_config.Bunch`

A class to define constant values for matrixAttributes of Matrix Routing API.

MatrixAttributes: `travelTimes`, `distances`.

`here_location_services.config.matrix_routing_config.MATRIX_ATTRIBUTES = {'distances': 'distances', 'travelTimes': 'travelTimes'}`

Use this config for matrix_attributes of matrix Routing API. Example: for travelTimes matrix_attributes use MATRIX_ATTRIBUTES.travelTimes.

```
class here_location_services.config.matrix_routing_config.AvoidFeatures(**kwargs)
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant values for features to avoid during matrix route calculation.

```
here_location_services.config.matrix_routing_config.AVOID_FEATURES =
{'controlledAccessHighway': 'controlledAccessHighway', 'dirtRoad': 'dirtRoad', 'ferry':
'ferry', 'tollRoad': 'tollRoad', 'tunnel': 'tunnel'}
    Use this config for avoid_features of matrix Routing API. Example: for tollRoad avoid_features use
    AVOID_FEATURES.tollRoad.
```

```
class here_location_services.config.matrix_routing_config.AvoidBoundingBox(north, south, west,
                                                                           east)
```

Bases: object

A class to define attributes of Avoid areaBoundingBox for Matrix Routing API.

Parameters

- **north** (*float*) –
- **south** (*float*) –
- **west** (*float*) –
- **east** (*float*) –

__init__ (*north, south, west, east*)

Parameters

- **north** (*float*) –
- **south** (*float*) –
- **west** (*float*) –
- **east** (*float*) –

__repr__ ()

Return string representation of this instance.

14.1.10 `here_location_services.config.search_config` module

This module defines all the configs which will be required as inputs to Geocoding and Search APIs.

```
class here_location_services.config.search_config.PlacesCategories(**kwargs)
    Bases: here\_location\_services.config.base\_config.Bunch
```

A class to define constant values for categories.

The HERE places category system provides three levels of granularity:

Level 1 - Categories that are organized into logical high-level groupings. For example, “100” represents the top level category for places associated with eating and drinking establishments (100=“Eat and Drink”).

Level 2 - Categories organized by logical sub-groups or domains. For example, “100-1000” represents the mid-level category for places associated with eating and drinking establishments that are restaurants (100-1000=“Eat and Drink / Restaurant”).

Level 3 - Categories that provide the greatest level of granularity about place categorization. For example, “100-1000-0001” is the complete category code for places associated with eating and drinking establishments that are casual dining restaurants (100-1000-0001=“Eat and Drink / Restaurant / Casual Dining”).

Detail information about each category is given [here](#).

```

here_location_services.config.search_config.PLACES_CATEGORIES = {'adult_entertainment':
'200-2000-0306', 'adult_shop': '600-6900-0305',
'advertising_marketing_pr_and_market_research': '700-7200-0252', 'aerial_tramway':
'400-4100-0340', 'airport': '400-4000-4581', 'airport_cargo': '400-4200-0052',
'airport_terminal': '400-4000-4582', 'ambulance_services': '700-7300-0280',
'amusement_park': '550-5520-0207', 'animal_park': '550-5520-0228',
'apartment_rental_flat_rental': '700-7200-0324', 'aquarium': '550-5520-0211',
'art_museum': '300-3100-0029', 'arts_and_crafts_supplies': '600-6900-0307', 'ashram':
'300-3200-0033', 'atm': '700-7010-0108', 'attorney': '700-7400-0138', 'auto_parts':
'700-7850-0123', 'automobile_club': '700-7850-0129', 'automobile_dealership_new_cars':
'700-7800-0118', 'automobile_dealership_used_cars': '700-7800-0119', 'aviation':
'700-7200-0330', 'b2b_restaurant_services': '700-7200-0329', 'b2b_sales_and_services':
'700-7200-0328', 'badminton': '800-8600-0200', 'bakery_and_baked_goods_store':
'600-6300-0244', 'bank': '700-7000-0107', 'banquet_hall': '800-8400-0139',
'bar_or_pub': '200-2000-0011', 'barber': '600-6950-0399', 'basketball':
'800-8600-0199', 'bay_harbor': '350-3500-0300', 'beach': '550-5510-0205',
'bed_and_breakfast': '500-5100-0058', 'beer_garden': '200-2000-0019',
'bicycle_and_bicycle_accessories_shop': '600-6900-0246', 'bicycle_parking':
'400-4100-0348', 'bicycle_service': '550-5510-0387', 'bicycle_service_and_maintenance':
'700-7400-0281', 'bicycle_sharing_location': '400-4100-0347', 'bike_park':
'800-8600-0376', 'bill_payment_service': '700-7400-0282', 'billiards_pool_hall':
'200-2000-0016', 'bistro': '100-1000-0008', 'blood_bank': '800-8000-0367', 'bmx_shop':
'600-6900-0394', 'bmx_track': '800-8600-0377', 'boat_ferry': '400-4100-0045',
'boating': '700-7400-0140', 'body_of_water': '350-3500-0233',
'body_piercing_and_tattoos': '700-7400-0283', 'bookstore': '600-6700-0087',
'border_crossing': '800-8100-0172', 'bowling_center': '800-8600-0184', 'brewery':
'300-3000-0350', 'building': '900-9300-0000', 'bus_rapid_transit': '400-4100-0341',
'bus_station': '400-4100-0036', 'bus_stop': '400-4100-0042', 'business_facility':
'700-7250-0136', 'business_service': '700-7400-0141', 'butcher': '600-6300-0363',
'cafeteria': '100-1000-0007', 'campground': '500-5100-0056', 'camping_hiking_shop':
'600-6900-0395', 'campsite': '550-5510-0378', 'canal': '350-3500-0303',
'canoe_kayak_shop': '600-6900-0396', 'car_repair': '700-7850-0122',
'car_repair_service': '700-7850-0000', 'car_wash_detailing': '700-7850-0121',
'cargo_center': '400-4200-0049', 'cargo_transportation': '400-4200-0241', 'casino':
'200-2300-0021', 'castle': '300-3000-0030', 'casual_dining': '100-1000-0001',
'catering_and_other_food_services': '700-7200-0253', 'cellphone_parking_lot':
'800-8500-0315', 'cemetery': '800-8700-0166', 'check_cashing_service_currency_exchange':
'700-7050-0110', 'childrens_apparel': '600-6800-0091', 'childrens_museum':
'300-3100-0027', 'chiropractor': '800-8000-0341', 'church': '300-3200-0030',
'cigar_and_tobacco_shop': '600-6900-0107', 'cinema': '200-2100-0019', 'city_hall':
'800-8100-0163', 'civic_community_center': '800-8100-0169', 'clothing_and_accessories':
'600-6800-0000', 'clubhouse': '800-8700-0296', 'coaching_institute': '800-8200-0360',
'cocktail_lounge': '200-2000-0368', 'coffee_shop': '100-1100-0010', 'coffee_tea':
'100-1100-0000', 'collective_community': '900-9200-0299', 'commercial_services':
'700-7200-0000', 'communication_media': '700-7100-0000', 'commuter_rail_station':
'400-4100-0038', 'commuter_train': '400-4100-0039', 'complete_rest_area':
'400-4300-0199', 'computer_and_software': '600-6500-0075', 'construction':
'700-7200-0254', 'consumer_electronics_store': '600-6500-0072', 'consumer_goods':
'600-6900-0000', 'consumer_services': '700-7400-0000', 'convenience_store':
'600-6000-0061', 'convention_exhibition_center': '800-8400-0176', 'county_council':
'800-8100-0168', 'couriers': '400-4200-0240', 'court_house': '800-8100-0170',
'covid_19_testing_site': '800-8000-0400', 'crematorium': '800-8700-0167',
'cross_country_ski_shop': '600-6900-0397', 'customer_care_service_center':
'700-7200-0255', 'dairy_goods': '600-6300-0364', 'dancing': '200-2000-0013', 'deli':
'100-1000-0006', 'delivery_entrance': '400-4200-0311', 'dentist_dental_office':
'800-8000-0154', 'department_store': '600-6200-0063', 'discount_store':
'600-6900-0100', 'distillery': '300-3000-0351', 'dining_center': '800-8000-0300',
'doughnut_shop': '600-6300-0246', 'drugstore': '600-6400-0069',
'drugstore_or_pharmacy': '600-6400-0000', 'dry_cleaning_and_laundry': '700-7400-0137',
'education_facility': '800-8200-0000', 'electrical': '700-7400-0365', 'embassy':

```

Use this config for categories of browse API. Example: for restaurant category use `PLACES_CATEGORIES.restaurant`.

14.1.11 `here_location_services.config.url_config` module

This module contains configurations for various Location services end points.

14.1.12 `here_location_services.autosuggest_api` module

This module contains classes for accessing [HERE Autosuggest API](#).

class `here_location_services.autosuggest_api.AutosuggestApi`(*api_key=None, auth=None, proxies=None, country='row'*)

Bases: `here_location_services.apis.Api`

A class for accessing HERE Autosuggest API.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

__init__(*api_key=None, auth=None, proxies=None, country='row'*)

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

get_autosuggest(*query, at=None, search_in_circle=None, search_in_bbox=None, in_country=None, limit=20, terms_limit=None, lang=None, political_view=None, show=None*)

Suggest address or place candidates based on an incomplete or misspelled query

Parameters

- **query** (*str*) – A string for free-text query. Example: res, rest
- **at** (*Optional[List]*) – Specify the center of the search context expressed as list of coordinates One of *at*, *search_in_circle* or *search_in_bbox* is required. Parameters “at”, “search_in_circle” and “search_in_bbox” are mutually exclusive. Only one of them is allowed.
- **search_in_circle** (*Optional[here_location_services.config.autosuggest_config.SearchCircle]*) – Search within a circular geographic area provided as latitude, longitude, and radius (in meters)
- **search_in_bbox** (*Optional[Tuple]*) – Search within a rectangular bounding box geographic area provided as tuple of west longitude, south latitude, east longitude, north latitude

- **in_country** (*Optional[List[str]]*) – Search within a specific or multiple countries provided as comma-separated ISO 3166-1 alpha-3 country codes. The country codes are to be provided in all uppercase. Must be accompanied by exactly one of *at*, *search_in_circle* or *search_in_bbox*.
- **limit** (*Optional[int]*) – An integer specifying maximum number of results to be returned.
- **terms_limit** (*Optional[int]*) – An integer specifying maximum number of Query Terms Suggestions to be returned.
- **lang** (*Optional[List[str]]*) – List of strings to select the language to be used for result rendering from a list of BCP 47 compliant language codes.
- **political_view** (*Optional[str]*) – Toggle the political view.
- **show** (*Optional[List[str]]*) – Select additional fields to be rendered in the response. Please note that some of the fields involve additional webservice calls and can increase the overall response time.

Returns requests.Response object.

Raises [ApiError](#) – If status_code of API response is not 200.

14.1.13 here_location_services.geocoding_search_api module

This module contains classes for accessing [HERE Geocoding & Search API](#).

```
class here_location_services.geocoding_search_api.GeocodingSearchApi(api_key=None,
                                                                    auth=None,
                                                                    proxies=None,
                                                                    country='row')
```

Bases: [here_location_services.apis.Api](#)

A class for accessing HERE Geocoding & search APIs.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

```
__init__(api_key=None, auth=None, proxies=None, country='row')
```

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

```
get_geocoding(query, limit=20, lang='en-US')
```

Get point for given free-form search query.

See further information here:

Parameters

- **query** (*str*) – a string containing the query to make.
- **limit** (*int*) – An int representing maximum number of results to be returned. Default value is 20.
- **lang** (*str*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `requests.Response` object.

Raises [`ApiError`](#) – If `status_code` of API response is not 200.

Return type `requests.models.Response`

get_reverse_geocoding(*lat, lng, limit=1, lang='en-US'*)

Get address for given latitude and longitude.

See further information here:

Parameters

- **lat** (*float*) – A float representing latitude of point.
- **lng** (*float*) – A float representing longitude of point.
- **limit** (*int*) – An int representing maximum number of results to be returned. Default value is 1.
- **lang** (*str*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `requests.Response` object.

Raises [`ApiError`](#) – If `status_code` of API response is not 200.

Return type `requests.models.Response`

get_search_discover(*query, center=None, radius=None, country_codes=None, bounding_box=None, limit=None, lang=None*)

Search places using Location Services discover endpoint.

This method uses location services `discover` endpoint to search places based on query which is free-form text.

Parameters

- **query** (*str*) – A string representing free-text query to search places.
- **center** (*Optional[List[float]]*) – A list of latitude and longitude representing the center for search query.
- **radius** (*Optional[int]*) – A radius in meters along with center for searching places.
- **country_codes** (*Optional[List]*) – A list of ISO 3166-1 alpha-3 country codes.
- **bounding_box** (*Optional[List[float]]*) – A bounding box, provided as west longitude, south latitude, east longitude, north latitude.
- **limit** (*Optional[int]*) – An int representing maximum number of results to be returned.
- **lang** (*Optional[str]*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `requests.Response` object.

Raises [`ApiError`](#) – If `status_code` of API response is not 200.

Return type `requests.models.Response`

get_search_browse(*center*, *radius=None*, *country_codes=None*, *bounding_box=None*, *categories=None*, *limit=None*, *name=None*, *lang=None*)

Get search results for places based on different filters such as categories or name.

Parameters

- **center** (*List*) – A list of latitude and longitude representing the center for search query.
- **radius** (*Optional[int]*) – A radius in meters along with center for searching places.
- **country_codes** (*Optional[List]*) – A list of ISO 3166-1 alpha-3 country codes.
- **bounding_box** (*Optional[List[float]]*) – A bounding box, provided as west longitude, south latitude, east longitude, north latitude.
- **categories** (*Optional[List]*) – A List of strings of category-ids.
- **limit** (*Optional[int]*) – An int representing maximum number of results to be returned.
- **name** (*Optional[str]*) – A string representing Full-text filter on POI names/titles.
- **lang** (*Optional[str]*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `requests.Response` object.

Raises [`ApiError`](#) – If `status_code` of API response is not 200.

Return type `requests.models.Response`

get_search_lookup(*location_id*, *lang=None*)

Get search results by providing `location_id`.

Parameters

- **location_id** (*str*) – A string representing id.
- **lang** (*Optional[str]*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `requests.Response` object.

Raises [`ApiError`](#) – If `status_code` of API response is not 200.

Return type `requests.models.Response`

14.1.14 here_location_services.isoline_routing_api module

This module contains classes for accessing [HERE Routing API](#).

class `here_location_services.isoline_routing_api.IsolineRoutingApi`(*api_key=None*, *auth=None*, *proxies=None*, *country='row'*)

Bases: `here_location_services.apis.Api`

A class for accessing HERE isoline routing API.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –

- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

`__init__(api_key=None, auth=None, proxies=None, country='row')`

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

get_isoline_routing(*range, range_type, transport_mode, origin=None, departure_time=None, destination=None, arrival_time=None, routing_mode='fast', shape_max_points=None, optimised_for='balanced', avoid_features=None, truck=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None*)

Get isoline routing.

Request a polyline that connects the endpoints of all routes leaving from one defined center with either a specified length or specified travel time.

Parameters

- **range** (*str*) – A string representing a range of isoline, unit is defined by parameter **range_type**. Example: `range='1000'` or `range='1000,2000,3000'`
- **range_type** (*str*) – A string representing a type of **range**. Possible values are **distance**, **time** and **consumption**. For distance the unit meters. For a time the unit is seconds. For consumption, it is defined by the consumption model.
- **transport_mode** (*str*) – A string representing Mode of transport to be used for the calculation of the isolines. Example: `car`.
- **origin** (*Optional[List]*) – Center of the isoline request. The Isoline(s) will cover the region which can be reached from this point within given range. It cannot be used in combination with **destination** parameter.
- **departure_time** (*Optional[datetime.datetime]*) – Specifies the time of departure as defined by either date-time or full-date partial-time in RFC 3339, section 5.6 (for example, `2019-06-24T01:23:45`). The requested time is converted to the local time at origin. When the optional timezone offset is not specified, time is assumed to be local. If neither **departure_time** or **arrival_time** are specified, current time at departure location will be used. All Time values in the response are returned in the timezone of each location.
- **destination** (*Optional[List]*) – Center of the isoline request. The Isoline(s) will cover the region within the specified range that can reach this point. It cannot be used in combination with **origin** parameter.
- **arrival_time** (*Optional[datetime.datetime]*) – Specifies the time of arrival as defined by either date-time or full-date T partial-time in RFC 3339, section 5.6 (for example, `2019-06-24T01:23:45`). The requested time is converted to the local time at destination. When the optional timezone offset is not specified, time is assumed to be local. All Time values in the response are returned in the timezone of each location.
- **routing_mode** (*Optional[str]*) – A string to represent routing mode.

- **shape_max_points** (*Optional[int]*) – An integer to Limit the number of points in the resulting isoline geometry. If the isoline consists of multiple components, the sum of points from all components is considered. This parameter doesn't affect performance.
- **optimised_for** (*Optional[str]*) – A string to specify how isoline calculation is optimized.
- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in :attr: `AVOID_FEATURES` <code>here_location_services.config.isoline_routing_config.AVOID_FEATURES</code>
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) – Different truck options to use during route calculation when transport_mode = truck. use object of Truck `here_location_services.config.base_config.Truck`
- **origin_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – PlaceOptions optional place options for origin.
- **origin_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – WayPointOptions optional waypoint options for origin.
- **destination_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – PlaceOptions optional place options for destination.
- **destination_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – WayPointOptions optional waypoint options for destination.

Returns requests.Response object.

Raises `ApiError` – If status_code of API response is not 200.

Return type requests.models.Response

14.1.15 here_location_services.matrix_routing_api module

This module contains classes for accessing HERE Matrix Routing API.

class `here_location_services.matrix_routing_api.MatrixRoutingApi`(*api_key=None, auth=None, proxies=None, country='row'*)

Bases: `here_location_services.apis.Api`

A class to access Matrix Routing API.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

__init__(*api_key=None, auth=None, proxies=None, country='row'*)

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –

- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

__send_post_request(*async_req, origins, region_definition, destinations=None, profile=None, departure_time=None, routing_mode=None, transport_mode=None, avoid_features=None, avoid_areas=None, truck=None, matrix_attributes=None*)

Parameters

- **async_req** (*str*) –
- **origins** (*List[Dict]*) –
- **region_definition** (*Union[here_location_services.config.matrix_routing_config.CircleRegion, here_location_services.config.matrix_routing_config.BoundingBoxRegion, here_location_services.config.matrix_routing_config.PolygonRegion, here_location_services.config.matrix_routing_config.AutoCircleRegion, here_location_services.config.matrix_routing_config.WorldRegion]*) –
- **destinations** (*Optional[List[Dict]]*) –
- **profile** (*Optional[str]*) –
- **departure_time** (*Optional[Union[datetime.datetime, str]]*) –
- **routing_mode** (*Optional[str]*) –
- **transport_mode** (*Optional[str]*) –
- **avoid_features** (*Optional[List[str]]*) –
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) –
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) –
- **matrix_attributes** (*Optional[List[str]]*) –

Return type Dict

matrix_route(*origins, region_definition, destinations=None, profile=None, departure_time=None, routing_mode=None, transport_mode=None, avoid_features=None, avoid_areas=None, truck=None, matrix_attributes=None*)

To Calculate routing matrix between multiple origins and destinations synchronously.

Parameters

- **origins** (*List[Dict]*) – A list of dictionaries containing lat and long for origin points.
- **region_definition** (*Union[here_location_services.config.matrix_routing_config.CircleRegion, here_location_services.config.matrix_routing_config.BoundingBoxRegion, here_location_services.config.matrix_routing_config.PolygonRegion, here_location_services.config.matrix_routing_config.AutoCircleRegion, here_location_services.config.matrix_routing_config.WorldRegion]*) – Definition of a region in which the matrix will be calculated. Use object of atleast one of the following regions: *here_location_services.config.matrix_routing_config.CircleRegion* *here_location_services.config.matrix_routing_config.BoundingBoxRegion* *here_location_services.config.matrix_routing_config.PolygonRegion* *here_location_services.config.matrix_routing_config.AutoCircleRegion* *here_location_services.config.matrix_routing_config.WorldRegion*

```
matrix_routing_config.AutoCircleRegion      here_location_services.  
config.matrix_routing_config.WorldRegion
```

- **destinations** (*Optional[List[Dict]]*) – A list of dictionaries containing lat and long for destination points. When no destinations are specified the matrix is assumed to be quadratic with origins used as destinations.
- **profile** (*Optional[str]*) – A string to represent profile id. A set predefined profile ids for route calculation can be used from config [PROFILE](#)
- **departure_time** (*Optional[Union[datetime.datetime, str]]*) – datetime.datetime object.
- **routing_mode** (*Optional[str]*) – A string to represent routing mode. Routing mode values are defined in [ROUTING_MODE](#)
- **transport_mode** (*Optional[str]*) – A string to represent transport mode. Transport modes are defined in [ROUTING_TRANSPORT_MODE](#)
- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area use object of `AvoidBoundingBox` `here_location_services.config.matrix_routing_config.AvoidBoundingBox`
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) – Different truck options to use during route calculation when `transportMode = truck`. use object of `Truck` `here_location_services.config.base_config.Truck`
- **matrix_attributes** (*Optional[List[str]]*) – Defines which attributes are included in the response as part of the data representation of the matrix entries summaries. Matrix attributes are defined in [MATRIX_ATTRIBUTES](#)

Returns `requests.Response` object.

Return type Dict

```
matrix_route_async(origins, region_definition, destinations=None, profile=None, departure_time=None,  
                    routing_mode=None, transport_mode=None, avoid_features=None,  
                    avoid_areas=None, truck=None, matrix_attributes=None)
```

To Calculate routing matrix between multiple origins and destinations asynchronously.

Parameters

- **origins** (*List[Dict]*) – A list of dictionaries containing lat and long for origin points.
- **region_definition** (*Union[here_location_services.config.matrix_routing_config.CircleRegion, here_location_services.config.matrix_routing_config.BoundingBoxRegion, here_location_services.config.matrix_routing_config.PolygonRegion, here_location_services.config.matrix_routing_config.AutoCircleRegion, here_location_services.config.matrix_routing_config.WorldRegion]*) – Definition of a region in which the matrix will be calculated. Use object of atleast one of the following regions: `here_location_services.config.matrix_routing_config.CircleRegion` `here_location_services.config.matrix_routing_config.BoundingBoxRegion` `here_location_services.config.matrix_routing_config.PolygonRegion` `here_location_services.config.`

```
matrix_routing_config.AutoCircleRegion      here_location_services.  
config.matrix_routing_config.WorldRegion
```

- **destinations** (*Optional[List[Dict]]*) – A list of dictionaries containing lat and long for destination points. When no destinations are specified the matrix is assumed to be quadratic with origins used as destinations.
- **profile** (*Optional[str]*) – A string to represent profile id. A set predefined profile ids for route calculation can be used from config [PROFILE](#)
- **departure_time** (*Optional[Union[datetime.datetime, str]]*) – datetime.datetime object.
- **routing_mode** (*Optional[str]*) – A string to represent routing mode. Routing mode values are defined in [ROUTING_MODE](#)
- **transport_mode** (*Optional[str]*) – A string to represent transport mode. Transport modes are defined in [ROUTING_TRANSPORT_MODE](#)
- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area use object of `AvoidBoundingBox` `here_location_services.config.matrix_routing_config.AvoidBoundingBox`
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) – Different truck options to use during route calculation when `transportMode = truck`. use object of `Truck` `here_location_services.config.matrix_routing_config.Truck`
- **matrix_attributes** (*Optional[List[str]]*) – Defines which attributes are included in the response as part of the data representation of the matrix entries summaries. Matrix attributes are defined in [MATRIX_ATTRIBUTES](#)

Returns `requests.Response` object.

Return type Dict

get_async_matrix_route_status(*status_url*)

Get the status of async matrix calculation for the provided status url.

Parameters **status_url** (*str*) –

Return type `requests.models.Response`

get_async_matrix_route_results(*result_url*)

Get the results of async matrix calculation for the provided result url.

Parameters **result_url** (*str*) –

Return type `requests.models.Response`

14.1.16 here_location_services.exceptions module

This module defines API exceptions.

exception here_location_services.exceptions.**ApiError**

Bases: Exception

Exception raised for API HTTP response status codes not in [200...300).

The exception value will be the response object returned by `requests` which provides access to all its attributes, eg. `status_code`, `reason` and `text`, etc.

__str__()

Return a string from the HTTP response causing the exception.

The string simply lists the response's status code, reason and text content, separated with commas.

exception here_location_services.exceptions.**ConfigException**

Bases: Exception

This ConfigException is raised whenever there is any error related to platform configuration.

exception here_location_services.exceptions.**AuthenticationException**(*resp*)

Bases: Exception

This AuthenticationException is raised either authentication or authorization on the platform fails.

__init__(*resp*)

Instantiate AuthenticationException . :param resp: response detail will be stored in this param

__str__()

Return the message to be raised for this exception.

Returns error message

Return type str

exception here_location_services.exceptions.**TooManyRequestsException**(*resp*)

Bases: Exception

Exception raised for API HTTP response status code 429.

The exception value will be the response object returned by `requests` which provides access to all its attributes, eg. `status_code`, `reason` and `text`, etc.

__init__(*resp*)

Instantiate AuthenticationException . :param resp: response detail will be stored in this param

__str__()

Return a string from the HTTP response causing the exception.

The string simply lists the response status code, reason and text content, separated with commas.

14.1.17 here_location_services.ls module

This module contains class to interact with Location services REST APIs.

```
class here_location_services.ls.LS(api_key=None, platform_credentials=None, proxies=None,  
                                country='row')
```

Bases: object

A single interface for the user to interact with rest of the Location services APIs.

Parameters

- **api_key** (*Optional[str]*) –
- **platform_credentials** (*Optional[here_location_services.platform.credentials.PlatformCredentials]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

```
__init__(api_key=None, platform_credentials=None, proxies=None, country='row')
```

Parameters

- **api_key** (*Optional[str]*) –
- **platform_credentials** (*Optional[here_location_services.platform.credentials.PlatformCredentials]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

```
geocode(query, limit=20, lang='en-US')
```

Calculate coordinates as result of geocoding for the given query.

Parameters

- **query** (*str*) – A string containing the input query.
- **limit** (*int*) – An int representing maximum number of results to be returned. Default value is 20.
- **lang** (*str*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Raises **ValueError** – If query is empty or having all whitespace characters.

Returns GeocoderResponse object.

Return type *here_location_services.responses.GeocoderResponse*

```
reverse_geocode(lat, lng, limit=1, lang='en-US')
```

Return the address label string as the result of reverse-geocoding the given latitude and longitude.

Parameters

- **lat** (*float*) – A float representing latitude of point.
- **lng** (*float*) – A float representing longitude of point.
- **limit** (*int*) – An int representing maximum number of results to be returned. Default value is 1.

- **lang** (*str*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Raises ValueError – If Latitude is not in range between -90 and 90 or Longitude is not in range between -180 and 180.

Returns ReverseGeocoderResponse object.

Return type *here_location_services.responses.ReverseGeocoderResponse*

calculate_isoline(*range, range_type, transport_mode, origin=None, departure_time=None, destination=None, arrival_time=None, routing_mode='fast', shape_max_points=None, optimised_for='balanced', avoid_features=None, truck=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None*)

Calculate isoline routing.

Request a polyline that connects the endpoints of all routes leaving from one defined center with either a specified length or specified travel time.

Parameters

- **range** (*str*) – A string representing a range of isoline, unit is defined by parameter range type. Example: range='1000' or range='1000,2000,3000'
- **range_type** (*str*) – A string representing a type of range. Possible values are distance, time and consumption. For distance the unit meters. For a time the unit is seconds. For consumption, it is defined by the consumption model.
- **transport_mode** (*str*) – A string representing Mode of transport to be used for the calculation of the isolines. Example: car.
- **origin** (*Optional[List]*) – Center of the isoline request. The Isoline(s) will cover the region which can be reached from this point within given range. It cannot be used in combination with destination parameter.
- **departure_time** (*Optional[datetime.datetime]*) – Specifies the time of departure as defined by either date-time or full-date partial-time in RFC 3339, section 5.6 (for example, 2019-06-24T01:23:45). The requested time is converted to the local time at origin. When the optional timezone offset is not specified, time is assumed to be local. If neither departure_time or arrival_time are specified, current time at departure location will be used. All Time values in the response are returned in the timezone of each location.
- **destination** (*Optional[List]*) – Center of the isoline request. The Isoline(s) will cover the region within the specified range that can reach this point. It cannot be used in combination with origin parameter.
- **arrival_time** (*Optional[datetime.datetime]*) – Specifies the time of arrival as defined by either date-time or full-date T partial-time in RFC 3339, section 5.6 (for example, 2019-06-24T01:23:45). The requested time is converted to the local time at destination. When the optional timezone offset is not specified, time is assumed to be local. All Time values in the response are returned in the timezone of each location.
- **routing_mode** (*Optional[str]*) – A string to represent routing mode.
- **shape_max_points** (*Optional[int]*) – An integer to Limit the number of points in the resulting isoline geometry. If the isoline consists of multiple components, the sum of points from all components is considered. This parameter doesn't affect performance.
- **optimised_for** (*Optional[str]*) – A string to specify how isoline calculation is optimized.

- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in :attr: `AVOID_FEATURES` <here_location_services.config.isoline_routing_config.AVOID_FEATURES>
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) – Different truck options to use during route calculation when `transport_mode = truck`. use object of `Truck` `here_location_services.config.base_config.Truck`
- **origin_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – `PlaceOptions` optional place options for origin.
- **origin_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – `WayPointOptions` optional waypoint options for origin.
- **destination_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – `PlaceOptions` optional place options for destination.
- **destination_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – `WayPointOptions` optional waypoint options for destination.

Raises `ValueError` – If origin and destination are provided together.

Returns `IsolineResponse` object.

Return type `here_location_services.responses.IsolineResponse`

autosuggest(*query, at=None, search_in_circle=None, search_in_bbox=None, in_country=None, limit=20, terms_limit=None, lang=None, political_view=None, show=None*)

Suggest address or place candidates based on an incomplete or misspelled query

Parameters

- **query** (*str*) – A string for free-text query. Example: `res, rest`
- **at** (*Optional[List]*) – Specify the center of the search context expressed as list of coordinates One of `at`, `search_in_circle` or `search_in_bbox` is required. Parameters “`at`”, “`search_in_circle`” and “`search_in_bbox`” are mutually exclusive. Only one of them is allowed.
- **search_in_circle** (*Optional[here_location_services.config.autosuggest_config.SearchCircle]*) – Search within a circular geographic area provided as latitude, longitude, and radius (in meters)
- **search_in_bbox** (*Optional[Tuple]*) – Search within a rectangular bounding box geographic area provided as tuple of west longitude, south latitude, east longitude, north latitude
- **in_country** (*Optional[List[str]]*) – Search within a specific or multiple countries provided as comma-separated ISO 3166-1 alpha-3 country codes. The country codes are to be provided in all uppercase. Must be accompanied by exactly one of `at`, `search_in_circle` or `search_in_bbox`.
- **limit** (*Optional[int]*) – An integer specifying maximum number of results to be returned.
- **terms_limit** (*Optional[int]*) – An integer specifying maximum number of Query Terms Suggestions to be returned.
- **lang** (*Optional[List[str]]*) – List of strings to select the language to be used for result rendering from a list of BCP 47 compliant language codes.

- **political_view** (*Optional[str]*) – Toggle the political view.
- **show** (*Optional[List[str]]*) – Select additional fields to be rendered in the response. Please note that some of the fields involve additional webservice calls and can increase the overall response time.

Returns requests.Response object.

Raises ValueError – If `search_in_circle`, `search_in_bbox` and `destination` are provided together.

Return type *here_location_services.responses.AutosuggestResponse*

get_dest_weather(*products*, *at=None*, *query=None*, *zipcode=None*, *hourly_date=None*,
one_observation=None, *language=None*, *units=None*)

Retrieves weather reports, weather forecasts, severe weather alerts and moon and sun rise and set information.

Parameters

- **products** (*List[str]*) – List of DestWeatherProduct identifying the type of report to obtain.
- **at** (*Optional[List]*) – A list of latitude and longitude specifying the area covered by the weather report.
- **query** (*Optional[str]*) – Free text query. Examples: “125, Berliner, berlin”, “Beacon, Boston”
- **zipcode** (*Optional[str]*) – ZIP code of the location. This parameter is supported only for locations in the United States of America.
- **hourly_date** (*Optional[Union[datetime.date, datetime.datetime]]*) – Date for which hourly forecasts are to be retrieved. Can be either a *date* or *datetime* object
- **one_observation** (*Optional[bool]*) – Boolean, if set to true, the response only includes the closest location. Only available when the *product* parameter is set to *DEST_WEATHER_PRODUCT.observation*.
- **language** (*Optional[str]*) – Defines the language used in the descriptions in the response.
- **units** (*Optional[str]*) – Defines whether units or imperial units are used in the response.

Raises

- **ValueError** – If neither *at*, *query* or *zipcode* are passed.
- **ValueError** – If *one_observation* is set to true without passing *DEST_WEATHER_PRODUCT.observation* in *products*

Returns DestinationWeatherResponse object.

Return type *here_location_services.responses.DestinationWeatherResponse*

get_weather_alerts(*geometry*, *start_time*, *id=None*, *weather_severity=None*, *weather_type=None*,
country=None, *end_time=None*, *width=None*)

Retrieves weather reports, weather forecasts, severe weather alerts and moon and sun rise and set information.

Parameters

- **geometry** (*Union[geojson.geometry.Point, geojson.geometry.LineString]*) – Point or LineString defining the route or a single location
- **start_time** (*datetime.datetime*) – Start time of the event
- **id** (*Optional[str]*) – Unique weather alert id.
- **weather_severity** (*Optional[int]*) – Defines the severity of the weather event as defined in `WeatherSeverity`.
- **weather_type** (*Optional[str]*) – Defines the type of the weather event as defined in `WeatherType`.
- **country** (*Optional[str]*) – String for ISO-3166-1 2-letter country code.
- **end_time** (*Optional[datetime.datetime]*) – End time of the event. If not present, warning is valid until it is not removed from the feed by national weather institutes (valid until warning is present in the response)
- **width** (*Optional[int]*) – int

Raises `ValueError` – If maximum width exceeds 100000 for point type geometry or width exceeds 25000 for LineString geometry

Returns `WeatherAlertsResponse` object.

Return type *here_location_services.responses.WeatherAlertsResponse*

solve_tour_planning(*fleet, plan, id=None, optimization_traffic=None, optimization_waiting_time=None, is_async=False*)

Requests profile-aware routing data, creates a Vehicle Routing Problem and solves it.

Parameters

- **fleet** (*here_location_services.config.tour_planning_config.Fleet*) – A fleet represented by various vehicle types for serving jobs.
- **plan** (*here_location_services.config.tour_planning_config.Plan*) – Represents the list of jobs to be served.
- **id** (*Optional[str]*) – A unique identifier of an entity. Avoid referencing any confidential or personal information as part of the Id.
- **optimization_traffic** (*Optional[str]*) – “liveOrHistorical” “historicalOnly” “automatic” Specifies what kind of traffic information should be considered for routing
- **optimization_waiting_time** (*Optional[Dict]*) – Configures departure time optimization which tries to adapt the starting time of the tour in order to reduce waiting time as a consequence of a vehicle arriving at a stop before the starting time of the time window defined for serving the job.
- **is_async** (*Optional[bool]*) – Solves the problem Asynchronously

Raises `ApiError` – If

Returns `TourPlanningResponse` object.

Return type *here_location_services.responses.TourPlanningResponse*

discover(*query, center=None, radius=None, country_codes=None, bounding_box=None, limit=None, lang=None*)

Search places using Location Services discover endpoint.

This method uses location services `discover` endpoint to search places based on query which is free-form text. There are three different combination of inputs as shown below to search places using `discover`:

- `center` and `country_code`
- `center` and `radius`
- `bounding_box`

Parameters

- **query** (*str*) – A string representing free-text query to search places.
- **center** (*Optional[List[float]]*) – A list of latitude and longitude representing the center for search query.
- **radius** (*Optional[int]*) – A radius in meters along with center for searching places.
- **country_codes** (*Optional[List]*) – A list of ISO 3166-1 alpha-3 country codes.
- **bounding_box** (*Optional[List[float]]*) – A bounding box, provided as west longitude, south latitude, east longitude, north latitude.
- **limit** (*Optional[int]*) – An int representing maximum number of results to be returned.
- **lang** (*Optional[str]*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Raises `ValueError` – If `center` and `bounding_box` are provided together.

Returns `DiscoverResponse` object.

Return type [*here_location_services.responses.DiscoverResponse*](#)

browse(*center, radius=None, country_codes=None, bounding_box=None, categories=None, limit=None, name=None, lang=None*)

Get search results for places based on different filters such as categories or name.

Parameters

- **center** (*List*) – A list of latitude and longitude representing the center for search query.
- **radius** (*Optional[int]*) – A radius in meters along with center for searching places.
- **country_codes** (*Optional[List]*) – A list of ISO 3166-1 alpha-3 country codes.
- **bounding_box** (*Optional[List[float]]*) – A bounding box, provided as west longitude, south latitude, east longitude, north latitude.
- **categories** (*Optional[List]*) – A list strings of category-ids.
- **limit** (*Optional[int]*) – An int representing maximum number of results to be returned.
- **name** (*Optional[str]*) – A string representing Full-text filter on POI names/titles.
- **lang** (*Optional[str]*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `BrowseResponse` object.

Return type [*here_location_services.responses.BrowseResponse*](#)

lookup(*location_id, lang=None*)

Get search results by providing `location_id`.

Parameters

- **location_id** (*str*) – A string representing id.
- **lang** (*Optional[[str](#)]*) – A string to represent language to be used for result rendering from a list of BCP47 compliant Language Codes.

Returns `LookupResponse` object.

Return type `here_location_services.responses.LookupResponse`

car_route(*origin, destination, via=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None, departure_time=None, routing_mode='fast', alternatives=0, units='metric', lang='en-US', return_results=None, spans=None, avoid_features=None, avoid_areas=None, exclude=None*)

Calculate car route between two endpoints.

Parameters

- **origin** (*List*) – A list of latitude and longitude of origin point of route.
- **destination** (*List*) – A list of latitude and longitude of destination point of route.
- **via** (*Optional[List[[here_location_services.config.routing_config.Via](#)]]*) – A list of Via objects.
- **origin_place_options** (*Optional[[here_location_services.config.base_config.PlaceOptions](#)]*) – PlaceOptions optional place options for origin.
- **origin_waypoint_options** (*Optional[[here_location_services.config.base_config.WayPointOptions](#)]*) – WayPointOptions optional waypoint options for origin.
- **destination_place_options** (*Optional[[here_location_services.config.base_config.PlaceOptions](#)]*) – PlaceOptions optional place options for destination.
- **destination_waypoint_options** (*Optional[[here_location_services.config.base_config.WayPointOptions](#)]*) – WayPointOptions optional waypoint options for destination.
- **departure_time** (*Optional[[datetime.datetime](#)]*) – `datetime.datetime` object.
- **routing_mode** (*str*) – A string to represent routing mode. use config defined in `ROUTING_MODE`
- **alternatives** (*int*) – Number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
- **units** (*str*) – A string representing units of measurement used in guidance instructions. The default is metric.
- **lang** (*str*) – A string representing preferred language of the response. The value should comply with the IETF BCP 47.
- **return_results** (*Optional[List]*) – A list of strings.
- **spans** (*Optional[List]*) – A list of strings to define which attributes are included in the response spans. use config defined in `ROUTING_SPANS`
- **avoid_features** (*Optional[List[[str](#)]]*) – Avoid routes that violate these properties. Avoid features are defined in `AVOID_FEATURES`
- **avoid_areas** (*Optional[List[[here_location_services.config.matrix_routing_config.AvoidBoundingBox](#)]]*) – A list of areas to avoid during route calculation. To define avoid area.

- **exclude** (*Optional[List[str]]*) – A comma separated list of three-letter country codes (ISO-3166-1 alpha-3 code) that routes will exclude.

Returns `RoutingResponse` object.

Return type `here_location_services.responses.RoutingResponse`

bicycle_route(*origin, destination, via=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None, departure_time=None, routing_mode='fast', alternatives=0, units='metric', lang='en-US', return_results=None, spans=None, avoid_features=None, avoid_areas=None, exclude=None*)

Calculate bicycle route between two endpoints.

Parameters

- **origin** (*List*) – A list of latitude and longitude of origin point of route.
- **destination** (*List*) – A list of latitude and longitude of destination point of route.
- **via** (*Optional[List[here_location_services.config.routing_config.Via]]*) – A list of `Via` objects.
- **origin_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – `PlaceOptions` optional place options for origin.
- **origin_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – `WayPointOptions` optional waypoint options for origin.
- **destination_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – `PlaceOptions` optional place options for destination.
- **destination_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – `WayPointOptions` optional waypoint options for destination.
- **departure_time** (*Optional[datetime.datetime]*) – `datetime.datetime` object.
- **routing_mode** (*str*) – A string to represent routing mode.
- **alternatives** (*int*) – Number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
- **units** (*str*) – A string representing units of measurement used in guidance instructions. The default is `metric`.
- **lang** (*str*) – A string representing preferred language of the response. The value should comply with the IETF BCP 47.
- **return_results** (*Optional[List]*) – A list of strings.
- **spans** (*Optional[List]*) – A list of strings to define which attributes are included in the response spans.
- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area.

- **exclude** (*Optional[List[str]]*) – A comma separated list of three-letter country codes (ISO-3166-1 alpha-3 code) that routes will exclude.

Returns `RoutingResponse` object.

Return type `here_location_services.responses.RoutingResponse`

truck_route(*origin, destination, via=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None, departure_time=None, routing_mode='fast', alternatives=0, units='metric', lang='en-US', return_results=None, spans=None, truck=None, avoid_features=None, avoid_areas=None, exclude=None*)

Calculate truck route between two endpoints.

Parameters

- **origin** (*List*) – A list of latitude and longitude of origin point of route.
- **destination** (*List*) – A list of latitude and longitude of destination point of route.
- **via** (*Optional[List[here_location_services.config.routing_config.Via]]*) – A list of `Via` objects.
- **origin_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – `PlaceOptions` optional place options for origin.
- **origin_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – `WayPointOptions` optional waypoint options for origin.
- **destination_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – `PlaceOptions` optional place options for destination.
- **destination_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – `WayPointOptions` optional waypoint options for destination.
- **departure_time** (*Optional[datetime.datetime]*) – `datetime.datetime` object.
- **routing_mode** (*str*) – A string to represent routing mode.
- **alternatives** (*int*) – Number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
- **units** (*str*) – A string representing units of measurement used in guidance instructions. The default is `metric`.
- **lang** (*str*) – A string representing preferred language of the response. The value should comply with the IETF BCP 47.
- **return_results** (*Optional[List]*) – A list of strings.
- **spans** (*Optional[List]*) – A list of strings to define which attributes are included in the response spans.
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) – Different truck options to use during route calculation. use object of `Truck` `here_location_services.config.matrix_routing_config.Truck`
- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)

- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area.
- **exclude** (*Optional[List[str]]*) – A comma separated list of three-letter country codes (ISO-3166-1 alpha-3 code) that routes will exclude.

Returns RoutingResponse object.

Return type *here_location_services.responses.RoutingResponse*

scooter_route(*origin, destination, via=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None, scooter=None, departure_time=None, routing_mode='fast', alternatives=0, units='metric', lang='en-US', return_results=None, spans=None, avoid_features=None, avoid_areas=None, exclude=None*)

Calculate scooter route between two endpoints.

Parameters

- **origin** (*List*) – A list of latitude and longitude of origin point of route.
- **destination** (*List*) – A list of latitude and longitude of destination point of route.
- **via** (*Optional[List[here_location_services.config.routing_config.Via]]*) – A list of Via objects.
- **origin_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – PlaceOptions optional place options for origin.
- **origin_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – WayPointOptions optional waypoint options for origin.
- **destination_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – PlaceOptions optional place options for destination.
- **destination_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – WayPointOptions optional waypoint options for destination.
- **scooter** (*Optional[here_location_services.config.routing_config.Scooter]*) – Additional attributes for scooter route.
- **departure_time** (*Optional[datetime.datetime]*) – datetime.datetime object.
- **routing_mode** (*str*) – A string to represent routing mode.
- **alternatives** (*int*) – Number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
- **units** (*str*) – A string representing units of measurement used in guidance instructions. The default is metric.
- **lang** (*str*) – A string representing preferred language of the response. The value should comply with the IETF BCP 47.
- **return_results** (*Optional[List]*) – A list of strings.
- **spans** (*Optional[List]*) – A list of strings to define which attributes are included in the response spans.

- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area.
- **exclude** (*Optional[List[str]]*) – A comma separated list of three-letter country codes (ISO-3166-1 alpha-3 code) that routes will exclude.

Returns `RoutingResponse` object.

Return type `here_location_services.responses.RoutingResponse`

pedestrian_route(*origin, destination, via=None, origin_place_options=None, origin_waypoint_options=None, destination_place_options=None, destination_waypoint_options=None, departure_time=None, routing_mode='fast', alternatives=0, units='metric', lang='en-US', return_results=None, spans=None, avoid_features=None, avoid_areas=None, exclude=None*)

Calculate pedestrian route between two endpoints.

Parameters

- **origin** (*List*) – A list of latitude and longitude of origin point of route.
- **destination** (*List*) – A list of latitude and longitude of destination point of route.
- **via** (*Optional[List[here_location_services.config.routing_config.Via]]*) – A list of Via objects.
- **origin_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – PlaceOptions optional place options for origin.
- **origin_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – WayPointOptions optional waypoint options for origin.
- **destination_place_options** (*Optional[here_location_services.config.base_config.PlaceOptions]*) – PlaceOptions optional place options for destination.
- **destination_waypoint_options** (*Optional[here_location_services.config.base_config.WayPointOptions]*) – WayPointOptions optional waypoint options for destination.
- **departure_time** (*Optional[datetime.datetime]*) – `datetime.datetime` object.
- **routing_mode** (*str*) – A string to represent routing mode.
- **alternatives** (*int*) – Number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
- **units** (*str*) – A string representing units of measurement used in guidance instructions. The default is metric.
- **lang** (*str*) – A string representing preferred language of the response. The value should comply with the IETF BCP 47.
- **return_results** (*Optional[List]*) – A list of strings.
- **spans** (*Optional[List]*) – A list of strings to define which attributes are included in the response spans.

- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area.
- **exclude** (*Optional[List[str]]*) – A comma separated list of three-letter country codes (ISO-3166-1 alpha-3 code) that routes will exclude.

Returns RoutingResponse object.

Return type [here_location_services.responses.RoutingResponse](#)

matrix(*origins, region_definition, async_req=False, destinations=None, profile=None, departure_time=None, routing_mode=None, transport_mode=None, avoid_features=None, avoid_areas=None, truck=None, matrix_attributes=None*)

Calculate routing matrix between multiple origins and destinations using synchronous and asynchronous requests.

A routing matrix is a matrix with rows labeled by origins and columns by destinations. Each entry of the matrix is travel time or distance from the origin to the destination. The response contains 2 optional flat arrays `TravelTimes` and `distances` depending upon the specified `matrix_attributes`. Each array represents a 2D matrix where rows (i) corresponds to origins and columns (j) to destinations. The kth position in the array corresponds to the (i, j) position in the matrix defined by the following relationship: $k = \text{num_destitions} * i + j$.

Parameters

- **origins** (*List[Dict]*) – A list of dictionaries containing lat and long for origin points.
- **region_definition** (*Union[here_location_services.config.matrix_routing_config.CircleRegion, here_location_services.config.matrix_routing_config.BoundingBoxRegion, here_location_services.config.matrix_routing_config.PolygonRegion, here_location_services.config.matrix_routing_config.AutoCircleRegion, here_location_services.config.matrix_routing_config.WorldRegion]*) – Definition of a region in which the matrix will be calculated. Use object of atleast one of the following regions:
 - [CircleRegion](#)
 - [BoundingBoxRegion](#)
 - [PolygonRegion](#)
 - [AutoCircleRegion](#)
 - [WorldRegion](#)
- **async_req** (*bool*) – If set to True reuquests will be sent to asynchronous matrix routing API else It will be sent to synchronous matrix routing API. For larger matrices, or longer routes, or routes in denser road networks, it is recommended to set to True.
- **destinations** (*Optional[List[Dict]]*) – A list of dictionaries containing lat and long for destination points. When no destinations are specified the matrix is assumed to be quadratic with origins used as destinations.
- **profile** (*Optional[str]*) – A string to represent profile id. A set predefined profile ids for route calculation can be used from config [PROFILE](#)

- **departure_time** (*Optional[Union[datetime.datetime, str]]*) – datetime.datetime object with explicit timezone. When departure_time is not specified, it is implicitly assumed to be the current time. The special value `any` enforces non time-aware routing.
- **routing_mode** (*Optional[str]*) – A string to represent routing mode. Routing mode values are defined in `ROUTING_MODE`
- **transport_mode** (*Optional[str]*) – A string to represent transport mode. Transport modes are defined in `ROUTING_TRANSPORT_MODE`
- **avoid_features** (*Optional[List[str]]*) – Avoid routes that violate these properties. Avoid features are defined in `AVOID_FEATURES`
- **avoid_areas** (*Optional[List[here_location_services.config.matrix_routing_config.AvoidBoundingBox]]*) – A list of areas to avoid during route calculation. To define avoid area use object of `AvoidBoundingBox`
- **truck** (*Optional[here_location_services.config.base_config.Truck]*) – Different truck options to use during route calculation when transport_mode = truck. use object of `Truck`
- **matrix_attributes** (*Optional[List[str]]*) – Defines which attributes are included in the response as part of the data representation of the matrix entries summaries. Matrix attributes are defined in `MATRIX_ATTRIBUTES`

Raises

- **ValueError** – If conflicting options are provided.
- **ApiError** – If API response status code is not as expected.

Returns `MatrixRoutingResponse` object.

Return type `here_location_services.responses.MatrixRoutingResponse`

14.1.18 here_location_services.responses module

This module contains classes for accessing the responses from Location Services RESTful APIs.

class `here_location_services.responses.ApiResponse(**kwargs)`

Bases: `object`

Base class for all the responses from Location Services RESTful APIs.

__init__(***kwargs*)

__str__()

Return `str(self)`.

as_json_string(*encoding='utf8'*)

Return API response as json string.

Parameters **encoding** (*str*) –

to_gejson()

Return API response as GeoJSON.

classmethod **new**(*resp*)

Instantiate a response object from raw response returned by API.

```
class here_location_services.responses.GeocoderResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the Geocoder API response data.
    __init__(**kwargs)

class here_location_services.responses.ReverseGeocoderResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the Reverse Geocoder API response data.
    __init__(**kwargs)

class here_location_services.responses.IsolineResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the Reverse Isoline routing API response data.
    __init__(**kwargs)
    to_geojson()
        Return API response as GeoJSON.

class here_location_services.responses.DiscoverResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the search discover API response data.
    __init__(**kwargs)

class here_location_services.responses.BrowseResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the search browse API response data.
    __init__(**kwargs)

class here_location_services.responses.LookupResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the search lookup API response data.
    __init__(**kwargs)

class here_location_services.responses.RoutingResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing the search routing API response data.
    __init__(**kwargs)
    to_geojson()
        Return API response as GeoJSON.

class here_location_services.responses.MatrixRoutingResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse
    A class representing Matrix routing response data.
    __init__(**kwargs)
    to_geojson()
        Return API response as GeoJSON.
    to_distnaces_matrix()
        Return distnaces matrix in a dataframe.
```

```

    to_travel_times_matrix()
        Return travel times matrix in a dataframe.

class here_location_services.responses.AutosuggestResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse

    A class representing the Autosuggest API response data.

    __init__(**kwargs)

class here_location_services.responses.DestinationWeatherResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse

    A class representing the Destination Weather API response data.

    __init__(**kwargs)

    to_geojson()
        Return API response as GeoJSON.

class here_location_services.responses.WeatherAlertsResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse

    A class representing the Destination Weather API response data.

    __init__(**kwargs)

    to_geojson()
        Return API response as GeoJSON.

class here_location_services.responses.TourPlanningResponse(**kwargs)
    Bases: here\_location\_services.responses.ApiResponse

    A class representing the Tour Planning API response data.

    __init__(**kwargs)

    to_geojson()
        Return API response as GeoJSON.

```

14.1.19 `here_location_services.routing_api` module

This module contains classes for accessing [HERE Routing API](#).

```

class here_location_services.routing_api.RoutingApi(api_key=None, auth=None, proxies=None,
                                                    country='row')

    Bases: here\_location\_services.apis.Api

    A class for accessing HERE routing APIs.

    Parameters
        • api_key (Optional[str]) –
        • auth (Optional[here\_location\_services.platform.auth.Auth]) –
        • proxies (Optional[dict]) –
        • country (str) –

    __init__(api_key=None, auth=None, proxies=None, country='row')

```

Parameters

- **api_key** (Optional[str]) –

- **auth** (*Optional* [`here_location_services.platform.auth.Auth`]) –
- **proxies** (*Optional* [`dict`]) –
- **country** (*str*) –

route(*transport_mode*, *origin*, *destination*, *via*=None, *origin_place_options*=None, *origin_waypoint_options*=None, *destination_place_options*=None, *destination_waypoint_options*=None, *scooter*=None, *departure_time*=None, *routing_mode*='fast', *alternatives*=0, *units*='metric', *lang*='en-US', *return_results*=None, *spans*=None, *truck*=None, *avoid_features*=None, *avoid_areas*=None, *exclude*=None)

Calculate route between two endpoints.

See further information *here* <<https://developer.here.com/documentation/routing-api/8.16.0/api-reference-swagger.html>>_.

Parameters

- **transport_mode** (*str*) – A string to represent mode of transport.
- **origin** (*List*) – A list of latitude and longitude of origin point of route.
- **destination** (*List*) – A list of latitude and longitude of destination point of route.
- **via** (*Optional* [*List* [`here_location_services.config.routing_config.Via`]]) – A list of tuples of latitude and longitude of via points.
- **origin_place_options** (*Optional* [`here_location_services.config.base_config.PlaceOptions`]) – PlaceOptions optional place options for origin.
- **origin_waypoint_options** (*Optional* [`here_location_services.config.base_config.WayPointOptions`]) – WayPointOptions optional waypoint options for origin.
- **destination_place_options** (*Optional* [`here_location_services.config.base_config.PlaceOptions`]) – PlaceOptions optional place options for destination.
- **destination_waypoint_options** (*Optional* [`here_location_services.config.base_config.WayPointOptions`]) – WayPointOptions optional waypoint options for destination.
- **scooter** (*Optional* [`here_location_services.config.routing_config.Scooter`]) – Additional attributes for scooter route.
- **departure_time** (*Optional* [`datetime.datetime`]) – `datetime.datetime` object.
- **routing_mode** (*str*) – A string to represent routing mode.
- **alternatives** (*int*) – Number of alternative routes to return aside from the optimal route. default value is 0 and maximum is 6.
- **units** (*str*) – A string representing units of measurement used in guidance instructions. The default is metric.
- **lang** (*str*) – A string representing preferred language of the response. The value should comply with the IETF BCP 47.
- **return_results** (*Optional* [*List*]) – A list of strings.
- **spans** (*Optional* [*List*]) – A list of strings to define which attributes are included in the response spans.

- **truck** (*Optional*[*here_location_services.config.base_config.Truck*])
– Different truck options to use during route calculation. use object of `Truck` `here_location_services.config.matrix_routing_config.Truck`
- **avoid_features** (*Optional*[*List*[*str*]]) – Avoid routes that violate these properties. Avoid features are defined in [AVOID_FEATURES](#)
- **avoid_areas** (*Optional*[*List*[*here_location_services.config.matrix_routing_config.AvoidBoundingBox*]]) – A list of areas to avoid during route calculation. To define avoid area use object of `AvoidBoundingBox` `here_location_services.config.matrix_routing_config.AvoidBoundingBox`.
- **exclude** (*Optional*[*List*[*str*]]) – A comma separated list of three-letter country codes (ISO-3166-1 alpha-3 code) that routes will exclude.

Returns `requests.Response` object.

Raises [ApiError](#) – If `status_code` of API response is not 200.

14.1.20 here_location_services.utils module

This is a collection of utilities for using Here Location Services.

`here_location_services.utils.get_apikey()`

Read and return the value of the environment variable `LS_API_KEY`.

Returns The string value of the environment variable or an empty string if no such variable could be found.

Return type `str`

14.1.21 here_location_services.platform package

Subpackages

`here_location_services.platform.apis` package

Submodules

`here_location_services.platform.apis.aaa_oauth2_api` module

This module contains an `AAA0auth2ApiClient` class to perform oauth API operations.

The HERE API reference documentation used in this module can be found [here](#):

class `here_location_services.platform.apis.aaa_oauth2_api.AAA0auth2Api` (*base_url*,
proxies=None)

Bases: `here_location_services.platform.apis.api.Api`

This class provides access to HERE platform AAA OAuth2 APIs.

Parameters

- **base_url** (*str*) –
- **proxies** (*Optional*[*dict*]) –

`__init__(base_url, proxies=None)`

Parameters

- **base_url** (*str*) –
- **proxies** (*Optional[dict]*) –

`request_scoped_access_token(oauth, data)`

Request scoped access oauth2 token from platform.

Parameters

- **oauth** (*requests_oauthlib.oauth1_auth.OAuth1*) – oauth1 configuration.
- **data** (*str*) – a string which represents request body.

Returns a json with scoped access token.

Return type Dict

`here_location_services.platform.apis.api` module

This module implements base class for low level api client.

`class here_location_services.platform.apis.api.Api(access_token, proxies=None)`

Bases: object

Base class for low level api calls.

Parameters **proxies** (*Optional[dict]*) –

`__init__(access_token, proxies=None)`

Parameters **proxies** (*Optional[dict]*) –

property headers: dict

Return HTTP request headers with Bearer token in Authorization field.

Returns authorization tokens

`post(url, data=None, params=None, headers=None, **kwargs)`

Perform a post request of an API at a specified URL with backoff.

Parameters

- **url** (*str*) – URL of the API.
- **data** (*Optional[Union[dict, list, bytes, str]]*) – Post data for http request.
- **params** (*Optional[dict]*) – Parameters to pass to the API.
- **headers** (*Optional[dict]*) – Request headers. Defaults to the api headers property.
- **kwargs** – Optional arguments that request takes.

Returns response from the API.

Return type requests.models.Response

`static raise_response_exception(resp)`

Parse HTTP errors status code and raise necessary exceptions.

Parameters **resp** (*requests.models.Response*) – An HTTP response to parse.

Raises

- **`TooManyRequestsException`** – If platform responds with HTTP 429.
- **`AuthenticationException`** – If platform responds with HTTP 401 or 403.
- **`Exception`** – If client responds with any other exception.

Return type None

Submodules**`here_location_services.platform.auth` module**

This module provides an `Auth` class to authenticate an app on the platform.

The authentication is based on some credentials object and will create an access token. It can be checked if the token is still valid, and it can be refreshed, too.

class `here_location_services.platform.auth.Auth(credentials, aaa_oauth2_api)`

Bases: object

This class is responsible for authenticating with the HERE platform.

It requires PlatformCredentials, AAAOAuth2BaseApi object.

Parameters

- **`credentials`** (`here_location_services.platform.credentials.PlatformCredentials`) –
- **`aaa_oauth2_api`** (`here_location_services.platform.apis.aaa_oauth2_api.AAAOAuth2Api`) –

`__init__`(`credentials, aaa_oauth2_api`)

Instantiate authentication token.

Parameters

- **`credentials`** (`here_location_services.platform.credentials.PlatformCredentials`) – an instance of PlatformCredentials
- **`aaa_oauth2_api`** (`here_location_services.platform.apis.aaa_oauth2_api.AAAOAuth2Api`) – an instance of AAAOAuth2Api required in case of Credentials type.

property `token`: **Optional[str]**

Return the current token or requests a new one if needed.

Returns a valid token

`token_still_valid()`

Check whether the auth token is still valid or expired.

Returns a boolean indicating if a token is still valid.

Return type bool

`generate_token()`

Authenticate with the HERE account service and retrieve a new token.

`here_location_services.platform.credentials` module

This module defines classes to manage Platform credentials.

class `here_location_services.platform.credentials.PlatformCredentials(cred_properties)`

Bases: object

Credentials provides functions for dealing with the HERE platform Credentials.

Credentials can be read from the following locations:

- The default location: “~/here/credentials.properties”
- A custom path to a credentials properties file
- Environment variables

Parameters `cred_properties` (*pyhocon.config_tree.ConfigTree*) –

__init__(*cred_properties*)

Instantiate the credentials object.

Parameters `cred_properties` (*pyhocon.config_tree.ConfigTree*) – the properties of Credentials.

classmethod `from_default()`

Return the credentials object from the default default credential path at ‘~/here/credentials.properties’.

If environmental variables are set, these values will override the ones found in the default file.

If no default file is found, this method will try to read the credentials from the environmental variables.

Returns credentials

Return type *here_location_services.platform.credentials.PlatformCredentials*

classmethod `from_credentials_file(path)`

Return the credentials object from a specified credentials path.

Parameters `path` (*Union[str, pathlib.Path]*) – path to a HERE platform credentials.properties file.

Returns credentials

Raises *ConfigException* – Erroneous credentials.properties file in path

Return type *here_location_services.platform.credentials.PlatformCredentials*

classmethod `from_env()`

Return the credentials object from the following environment variables:

- HERE_USER_ID
- HERE_CLIENT_ID
- HERE_ACCESS_KEY_ID
- HERE_ACCESS_KEY_SECRET
- HERE_TOKEN_ENDPOINT_URL (optional)

Returns credentials parsed from the environment variables

Raises *ConfigException* – missing environmental variables that are mandatory

Return type *here_location_services.platform.credentials.PlatformCredentials*

patch_using_env()

Patch the credentials by reading the following environment variables and applying them accordingly.

- `HERE_USER_ID`
- `HERE_CLIENT_ID`
- `HERE_ACCESS_KEY_ID`
- `HERE_ACCESS_KEY_SECRET`
- `HERE_TOKEN_ENDPOINT_URL`

Whenever such an environment variable is set, it overrides the one loaded from file.

14.1.22 here_location_services.destination_weather_api module

This module contains classes for accessing [HERE Destination Weather API](#).

class `here_location_services.destination_weather_api.DestinationWeatherApi`(*api_key=None, auth=None, proxies=None, country='row'*)

Bases: [here_location_services.apis.Api](#)

A class for accessing HERE routing APIs.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

__init__(*api_key=None, auth=None, proxies=None, country='row'*)

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

get_dest_weather(*products, at=None, query=None, zipcode=None, hourly_date=None, one_observation=None, language=None, units=None*)

Retrieves weather reports, weather forecasts, severe weather alerts and moon and sun rise and set information.

See further information [Here Destination Weather API <https://developer.here.com/documentation/destination-weather/dev_guide/topics/overview.html>](https://developer.here.com/documentation/destination-weather/dev_guide/topics/overview.html).

Parameters

- **products** (*List[str]*) – List of DestWeatherProduct identifying the type of report to obtain.
- **at** (*Optional[List]*) – A list of latitude and longitude specifying the area covered by the weather report.

- **query** (*Optional[str]*) – Free text query. Examples: “125, Berliner, berlin”, “Beacon, Boston”
- **zipcode** (*Optional[str]*) – ZIP code of the location. This parameter is supported only for locations in the United States of America.
- **hourly_date** (*Optional[Union[datetime.date, datetime.datetime]]*) – Date for which hourly forecasts are to be retrieved. Can be either a *date* or *datetime* object
- **one_observation** (*Optional[bool]*) – Boolean, if set to true, the response only includes the closest location. Only available when the *product* parameter is set to *DEST_WEATHER_PRODUCT.observation*.
- **language** (*Optional[str]*) – Defines the language used in the descriptions in the response.
- **units** (*Optional[str]*) – Defines whether units or imperial units are used in the response.

Returns requests.Response object.

Raises *ApiError* – If status_code of API response is not 200.

get_weather_alerts(*geometry, start_time, id=None, weather_severity=None, weather_type=None, country=None, end_time=None, width=None*)

Retrieves weather reports, weather forecasts, severe weather alerts and moon and sun rise and set information.

See further information *Here Destination Weather API* <https://developer.here.com/documentation/destination-weather/dev_guide/topics/overview.html>_.

Parameters

- **geometry** (*Union[geojson.geometry.Point, geojson.geometry.LineString]*) – Point or LineString defining the route or a single location
- **start_time** (*datetime.datetime*) – Start time of the event
- **id** (*Optional[str]*) – Unique weather alert id.
- **weather_severity** (*Optional[int]*) – Defines the severity of the weather event as defined in *WeatherSeverity*.
- **weather_type** (*Optional[str]*) – Defines the type of the weather event as defined in *WeatherType*.
- **country** (*Optional[str]*) – String for ISO-3166-1 2-letter country code.
- **end_time** (*Optional[datetime.datetime]*) – End time of the event. If not present, warning is valid until it is not removed from the feed by national weather institutes (valid until warning is present in the response)
- **width** (*Optional[int]*) – int

Returns requests.Response object.

Raises *ApiError* – If status_code of API response is not 200.

14.1.23 here_location_services.config.tour_planning_config module

This module defines all the configs which will be required as inputs to Tour planning API.

class here_location_services.config.tour_planning_config.TourPlanningAvoidFeatures(**kwargs)
 Bases: [here_location_services.config.base_config.Bunch](#)

A class to define values for features to avoid features during tour planning calculation.

here_location_services.config.tour_planning_config.TOUR_PLANNING_AVOID_FEATURES =
 {'dirtRoad': 'dirtRoad', 'ferry': 'ferry', 'motorway': 'motorway', 'tollRoad':
 'tollRoad', 'tunnel': 'tunnel'}

Use this config for avoid of Tour planning API. Example: for tollRoad avoids use
 TOUR_PLANNING_AVOID_FEATURES.tollRoad.

class here_location_services.config.tour_planning_config.VehicleMode(**kwargs)
 Bases: [here_location_services.config.base_config.Bunch](#)

A class to define values vehicle mode in vehicle profile.

here_location_services.config.tour_planning_config.VEHICLE_MODE = {'bicycle': 'bicycle',
 'car': 'car', 'pedestrian': 'pedestrian', 'scooter': 'scooter', 'truck': 'truck'}

Use this config for vehicle_mode of Tour planning API. Example: for scooter avoids use VEHICLE_MODE.
 scooter.

class here_location_services.config.tour_planning_config.VehicleType(id, profile_name, amount,
 capacity, shift_start,
 shift_end=None,
 shift_breaks=None,
 costs_fixed=0,
 costs_distance=0,
 costs_time=0,
 skills=None,
 limits=None)

Bases: object

A class to define VehicleType

Type of vehicle in a fleet

Parameters

- **id** (str) –
- **profile_name** (str) –
- **amount** (int) –
- **capacity** (List[int]) –
- **shift_start** (Dict) –
- **shift_end** (Optional[Dict]) –
- **shift_breaks** (Optional[List]) –
- **costs_fixed** (float) –
- **costs_distance** (float) –
- **costs_time** (float) –
- **skills** (Optional[List[str]]) –
- **limits** (Optional[Dict]) –

```
__init__(id, profile_name, amount, capacity, shift_start, shift_end=None, shift_breaks=None,
         costs_fixed=0, costs_distance=0, costs_time=0, skills=None, limits=None)
```

Parameters

- **id** (*str*) – Specifies id of the vehicle type. Avoid assigning real-life identifiers, such as vehicle license plate as the id of a vehicle
- **profile_name** (*str*) – characters `^[a-zA-Z0-9_-]+$` Specifies the name of the profile. Avoid assigning real-life identifiers, such as a vehicle license plate Id or personal name as the `profileName` of the routing profile.
- **amount** (*int*) – Amount of vehicles available.
- **capacity** (*List[int]*) – Unit of measure, e.g. volume, mass, size, etc.
- **shift_start** (*Dict*) – Represents a depot: a place where a vehicle starts
- **costs_fixed** (*float*) – A fixed cost to start using vehicle of this type. It is optional with a default value of zero
- **shift_end** (*Optional[Dict]*) – Represents a depot: a place where a vehicle ends
- **shift_breaks** (*Optional[List]*) – Represents a depot: a place where a vehicle takes breaks
- **costs_distance** (*float*) – A cost per meter. It is optional with a default value of zero.
- **costs_time** (*float*) – A cost per second. It is optional with a default value of zero. In case time and distance costs are zero then a small time cost 0.000000000001 will be used instead
- **skills** (*Optional[List[str]]*) – A list of skills for a vehicle or a job.
- **limits** (*Optional[Dict]*) – Contains constraints applied to a vehicle type.

```
class here_location_services.config.tour_planning_config.VehicleProfile(name, vehicle_mode,
                                                                       departure_time=None,
                                                                       avoid=None,
                                                                       truck_options=None,
                                                                       al-
                                                                       low_highway_for_scooter=None)
```

Bases: object

A class to define VehicleProfile

Profile of vehicle in a fleet

Parameters

- **name** (*str*) – Specifies the name of the profile. Avoid assigning real-life identifiers, such as a vehicle license plate Id or personal name as the `profileName` of the routing profile.
- **vehicle_mode** (*str*) – Contains constraints applied to a vehicle type.
- **departure_time** (*Optional[datetime.datetime]*) – Represents time of departure.
- **avoid** (*Optional[List[here_location_services.config.tour_planning_config.TourPlanningAvoidFeatures]]*) – Avoid routes that violate these properties.
- **truck_options** (*Optional[here_location_services.config.base_config.Truck]*) – Specifies truck profile options.

- **allow_highway_for_scooter** (*Optional[bool]*) – Specifies whether routing calculation should take highways into account. When this parameter isn't provided, then by default highways would be avoided. If the avoid feature motorway is provided, then highways would be avoided, even if this is set to true.

Raises

- **ValueError** – If `truck_options` are provided without setting `vehicle_mode` to `VEHICLE_MODE.truck`
- **ValueError** – If `allow_highway_for_scooter` is provided without setting `vehicle_mode` to `VEHICLE_MODE.scooter`

__init__ (*name, vehicle_mode, departure_time=None, avoid=None, truck_options=None, allow_highway_for_scooter=None*)

Parameters

- **name** (*str*) –
- **vehicle_mode** (*str*) –
- **departure_time** (*Optional[datetime.datetime]*) –
- **avoid** (*Optional[List[here_location_services.config.tour_planning_config.TourPlanningAvoidFeatures]]*) –
- **truck_options** (*Optional[here_location_services.config.base_config.Truck]*) –
- **allow_highway_for_scooter** (*Optional[bool]*) –

class `here_location_services.config.tour_planning_config.Fleet` (*vehicle_types, vehicle_profiles*)

Bases: `object`

A class to define Fleet

A fleet represented by various vehicle types for serving jobs.

Parameters

- **vehicle_types** (*List[here_location_services.config.tour_planning_config.VehicleType]*) – A list of vehicle types. The upper limit for the number of vehicle types is 35 for the synchronous problems endpoint and 150 for the asynchronous problems endpoint.
- **vehicle_profiles** (*List*) – Specifies the profile of the vehicle.

__init__ (*vehicle_types, vehicle_profiles*)

Parameters

- **vehicle_types** (*List[here_location_services.config.tour_planning_config.VehicleType]*) –
- **vehicle_profiles** (*List*) –

class `here_location_services.config.tour_planning_config.JobPlaces` (*duration, demand, location, tag=None, times=None*)

Bases: `object`

A class to define JobPlaces

Parameters

- **duration** (*int*) – Represents duration in seconds.
- **demand** (*List[int]*) – Unit of measure, e.g. volume, mass, size, etc.
- **location** (*Tuple*) – Represents geospatial location defined by latitude and longitude.
- **tag** (*Optional[str]*) – A free text associated with the job place. Avoid referencing any confidential or personal information as part of the JobTag.
- **times** (*Optional[List[List[str]]]*) – Represents multiple time windows.

__init__ (*duration, demand, location, tag=None, times=None*)

Parameters

- **duration** (*int*) –
- **demand** (*List[int]*) –
- **location** (*Tuple*) –
- **tag** (*Optional[str]*) –
- **times** (*Optional[List[List[str]]]*) –

```
class here_location_services.config.tour_planning_config.Job(id, skills=None, priority=None, pickups=None, deliveries=None)
```

Bases: object

A class to define Job

A fleet represented by various vehicle types for serving jobs.

Parameters

- **id** (*str*) – Specifies id of the job. Avoid referencing any sensitive or personal information, such as names, addresses, information about a delivery or service, as part of the jobId.
- **skills** (*Optional[List]*) – A list of skills for a vehicle or a job.
- **priority** (*Optional[int]*) – Specifies the priority of the job with 1 for high priority jobs and 2 for normal jobs.
- **pickups** (*Optional[List[here_location_services.config.tour_planning_config.JobPlaces]]*) – Places where sub jobs to be performed. All pickups are done before any other delivery.
- **deliveries** (*Optional[List[here_location_services.config.tour_planning_config.JobPlaces]]*) – Places where sub jobs to be performed. All pickups are done before any other delivery.

Raises ValueError – If no subjob is specified either as pickup or delivery.

__init__ (*id, skills=None, priority=None, pickups=None, deliveries=None*)

Parameters

- **id** (*str*) –
- **skills** (*Optional[List]*) –
- **priority** (*Optional[int]*) –
- **pickups** (*Optional[List[here_location_services.config.tour_planning_config.JobPlaces]]*) –

- **deliveries** (Optional[List[here_location_services.config.tour_planning_config.JobPlaces]]) –

class here_location_services.config.tour_planning_config.**Relation**(type, jobs, vehicle_id)

Bases: object

A class to define Relation

Represents a list of preferred relations between jobs, vehicles.

Parameters

- **type** (str) – “sequence” “tour” “flexible” Defines a relation between jobs and a specific vehicle
- **jobs** (List) – Ids of jobs or reserved activities. There are three reserved activity ids: - departure: specifies departure activity. Should be first in the list. - break: specifies vehicle break activity - arrival: specifies arrival activity. Should be last in the list.
- **vehicle_id** (str) – A unique identifier of an entity. Avoid referencing any confidential or personal information as part of the Id.

__init__(type, jobs, vehicle_id)

Parameters

- **type** (str) –
- **jobs** (List) –
- **vehicle_id** (str) –

class here_location_services.config.tour_planning_config.**Plan**(jobs, relations=None)

Bases: object

A class to define Plan

Represents the list of jobs to be served.

Parameters

- **jobs** (List[here_location_services.config.tour_planning_config.Job]) –
- **relations** (Optional[List[here_location_services.config.tour_planning_config.Relation]]) –

__init__(jobs, relations=None)

Parameters

- **jobs** (List[here_location_services.config.tour_planning_config.Job]) –
- **relations** (Optional[List[here_location_services.config.tour_planning_config.Relation]]) –

14.1.24 here_location_services.tour_planning_api module

This module contains classes for accessing [HERE Tour Planning API](#).

```
class here_location_services.tour_planning_api.TourPlanningApi(api_key=None, auth=None,  
                                                             proxies=None, country='row')
```

Bases: [here_location_services.apis.Api](#)

A class for accessing HERE Tour Planning API.

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

```
__init__(api_key=None, auth=None, proxies=None, country='row')
```

Parameters

- **api_key** (*Optional[str]*) –
- **auth** (*Optional[here_location_services.platform.auth.Auth]*) –
- **proxies** (*Optional[dict]*) –
- **country** (*str*) –

```
solve_tour_planning(fleet, plan, id=None, optimization_traffic=None, optimization_waiting_time=None,  
                   is_async=False)
```

Requests profile-aware routing data, creates a Vehicle Routing Problem and solves it.

Parameters

- **fleet** ([here_location_services.config.tour_planning_config.Fleet](#)) – A fleet represented by various vehicle types for serving jobs.
- **plan** ([here_location_services.config.tour_planning_config.Plan](#)) – Represents the list of jobs to be served.
- **id** (*Optional[str]*) – A unique identifier of an entity. Avoid referencing any confidential or personal information as part of the Id.
- **optimization_traffic** (*Optional[str]*) – “liveOrHistorical” “historicalOnly” “automatic” Specifies what kind of traffic information should be considered for routing
- **optimization_waiting_time** (*Optional[Dict]*) – Configures departure time optimization which tries to adapt the starting time of the tour in order to reduce waiting time as a consequence of a vehicle arriving at a stop before the starting time of the time window defined for serving the job.
- **is_async** (*Optional[bool]*) – Solves the problem Asynchronously

Returns requests.Response object.

Raises [ApiError](#) – If status_code of API response is not 200 or 202.

```
get_async_tour_planning_status(status_url)
```

Get the status of async tour planning calculation for the provided status url.

Parameters **status_url** (*str*) –

Return type requests.models.Response

get_async_tour_planning_results(*result_url*)

Get the results of async tour planning for the provided result url.

Parameters **result_url** (*str*) –

CHANGELOG

15.1 here-location-services 0.4.0 (2021-09-07)

- Added Destination Weather API
- Added Autosuggest API
- Updated Isoline Routing API from v7 to v8
- Added support for authentication token from HERE platform

15.2 here-location-services 0.3.0 (2021-07-22)

- Fixed Via waypoint and place options for routing.
- Added interactive examples in the documentation.

15.3 here-location-services 0.2.0 (2021-04-19)

- Add Routing API
- Add Matrix Routing API
- Add config for categories

15.4 here-location-services 0.1.0 (2021-02-15)

- Initial release

CONTRIBUTING TO HERE LOCATION SERVICES FOR PYTHON

Thank you for taking the time to contribute.

The following is a set of guidelines for contributing to this package. These are mostly guidelines, not rules. Use your best judgement and feel free to propose changes to this document in a pull request.

16.1 Coding Guidelines

- Lint your code contributions as per [pep8 guidelines](#).

To help you out, we have included a *Makefile* in the root directory which supports the commands below: Auto-format code using black:

```
make black
```

Check for linting errors:

```
make lint
```

- Sort the imports in each python file as per [pep8 guidelines imports](#). Please execute the isort utility to have the imports sorted auto-magically.

16.2 Notebooks

Example Notebooks are provided in [Notebooks](#).

16.3 Signing each Commit

When you file a pull request, we ask that you sign off the [Developer Certificate of Origin](#) (DCO) in each commit. Any Pull Request with commits that are not signed off will be rejected by the [DCO check](#).

A DCO is a lightweight way to confirm that a contributor wrote or otherwise has the right to submit code or documentation to a project. Simply add Signed-off-by as shown in the example below to indicate that you agree with the DCO.

The git flag `-s` can be used to sign a commit:

```
git commit -s -m 'README.md: Fix minor spelling mistake'
```

The result is a signed commit message:

README.md: Fix minor spelling mistake

Signed-off-by: John Doe <john.doe@example.com>

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

h here_location_services.utils, 89

here_location_services, 47

here_location_services.__version__, 47

here_location_services.apis, 47

here_location_services.autosuggest_api, 63

here_location_services.config, 48

here_location_services.config.autosuggest_config, 51

here_location_services.config.base_config, 48

here_location_services.config.dest_weather_config, 53

here_location_services.config.isoline_routing_config, 52

here_location_services.config.matrix_routing_config, 58

here_location_services.config.routing_config, 55

here_location_services.config.search_config, 60

here_location_services.config.tour_planning_config, 95

here_location_services.config.url_config, 63

here_location_services.destination_weather_api, 93

here_location_services.exceptions, 72

here_location_services.geocoding_search_api, 64

here_location_services.isoline_routing_api, 66

here_location_services.ls, 73

here_location_services.matrix_routing_api, 68

here_location_services.platform, 89

here_location_services.platform.apis, 89

here_location_services.platform.apis.aaa_oauth2_api, 89

here_location_services.platform.apis.api, 90

here_location_services.platform.auth, 91

here_location_services.platform.credentials, 92

here_location_services.responses, 85

here_location_services.routing_api, 87

here_location_services.tour_planning_api, 100

Symbols

<code>__add_api_key_in_params()</code> (<i>here_location_services.apis.Api</i> method), 48	<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.Relation</i> method), 99
<code>__init__()</code> (<i>here_location_services.apis.Api</i> method), 47	<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.VehicleP</i> method), 97
<code>__init__()</code> (<i>here_location_services.autosuggest_api.AutosuggestApi</i> method), 63	<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.VehicleT</i> method), 95
<code>__init__()</code> (<i>here_location_services.config.autosuggest_config.SearchCircle</i> method), 52	<code>__init__()</code> (<i>here_location_services.destination_weather_api.Destination</i> method), 93
<code>__init__()</code> (<i>here_location_services.config.base_config.Bunch</i> method), 48	<code>__init__()</code> (<i>here_location_services.exceptions.AuthenticationException</i> method), 72
<code>__init__()</code> (<i>here_location_services.config.base_config.PlaceOptions</i> method), 50	<code>__init__()</code> (<i>here_location_services.exceptions.TooManyRequestsExceptio</i> method), 72
<code>__init__()</code> (<i>here_location_services.config.base_config.Truck</i> method), 49	<code>__init__()</code> (<i>here_location_services.geocoding_search_api.GeocodingSea</i> method), 64
<code>__init__()</code> (<i>here_location_services.config.base_config.WayPointOptions</i> method), 51	<code>__init__()</code> (<i>here_location_services.isoline_routing_api.IsolineRoutingAp</i> method), 67
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.AutoCircleRegion</i> method), 59	<code>__init__()</code> (<i>here_location_services.ls.LS</i> method), 73
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.AvoidBoundingBox</i> method), 60	<code>__init__()</code> (<i>here_location_services.matrix_routing_api.MatrixRoutingAp</i> method), 68
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.BoundingBoxRegion</i> method), 58	<code>__init__()</code> (<i>here_location_services.platform.apis.aaa_oauth2_api.AAAO</i> method), 89
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.CircleRegion</i> method), 58	<code>__init__()</code> (<i>here_location_services.platform.apis.api.Api</i> method), 90
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.PolygonRegion</i> method), 58	<code>__init__()</code> (<i>here_location_services.platform.auth.Auth</i> method), 91
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.PolylineRegion</i> method), 58	<code>__init__()</code> (<i>here_location_services.platform.credentials.PlatformCreden</i> method), 92
<code>__init__()</code> (<i>here_location_services.config.matrix_routing_config.WorldRegion</i> method), 59	<code>__init__()</code> (<i>here_location_services.responses.ApiResponse</i> method), 85
<code>__init__()</code> (<i>here_location_services.config.routing_config.Scooter</i> method), 57	<code>__init__()</code> (<i>here_location_services.responses.AutosuggestResponse</i> method), 87
<code>__init__()</code> (<i>here_location_services.config.routing_config.Via</i> method), 57	<code>__init__()</code> (<i>here_location_services.responses.BrowseResponse</i> method), 86
<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.Fleet</i> method), 97	<code>__init__()</code> (<i>here_location_services.responses.DestinationWeatherRespon</i> method), 87
<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.Job</i> method), 98	<code>__init__()</code> (<i>here_location_services.responses.DiscoverResponse</i> method), 86
<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.JobPluces</i> method), 98	<code>__init__()</code> (<i>here_location_services.responses.GeocoderResponse</i> method), 86
<code>__init__()</code> (<i>here_location_services.config.tour_planning_config.Plan</i> method), 99	<code>__init__()</code> (<i>here_location_services.responses.IsolineResponse</i> method), 86
	<code>__init__()</code> (<i>here_location_services.responses.LookupResponse</i> method), 86

method), 86
 __init__() (here_location_services.responses.MatrixRoutingResponse), 85
 method), 86
 __init__() (here_location_services.responses.ReverseGeocoderResponse), 85
 method), 86
 __init__() (here_location_services.responses.RoutingResponse), 86
 method), 86
 __init__() (here_location_services.responses.TourPlanningResponse), 87
 method), 87
 __init__() (here_location_services.responses.WeatherAlertResponse), 75
 method), 87
 __init__() (here_location_services.routing_api.RoutingApi), 87
 method), 87
 __init__() (here_location_services.tour_planning_api.TourPlanningApi), 100
 method), 100
 __repr__() (here_location_services.config.base_config.PlaceOptions), 51
 method), 51
 __repr__() (here_location_services.config.base_config.WayPointOptions), 51
 method), 51
 __repr__() (here_location_services.config.matrix_routing_config.AvoidFeaturesRegion), 59
 method), 59
 __repr__() (here_location_services.config.matrix_routing_config.AvoidBoundingBox), 60
 method), 60
 __repr__() (here_location_services.config.matrix_routing_config.AvoidBoundedRegion), 58
 method), 58
 __repr__() (here_location_services.config.matrix_routing_config.AvoidCircleRegion), 58
 method), 58
 __repr__() (here_location_services.config.matrix_routing_config.PolygonRegion), 59
 method), 59
 __repr__() (here_location_services.config.routing_config.Scooter), 57
 method), 57
 __send_post_request()
 (here_location_services.matrix_routing_api.MatrixRoutingApi
 method), 69
 __str__() (here_location_services.exceptions.ApiError), 72
 method), 72
 __str__() (here_location_services.exceptions.AuthenticationException), 72
 method), 72
 __str__() (here_location_services.exceptions.TooManyRequestsException), 72
 method), 72
 __str__() (here_location_services.responses.ApiResponse), 85
 method), 85
 _get_url_string() (here_location_services.apis.Api
 method), 47

A
 AAOAuth2Api (class in here_location_services.platform.apis.aaa_oauth2_api), 89
 Api (class in here_location_services.apis), 47
 Api (class in here_location_services.platform.apis.api), 90
 ApiError, 72

ApiResponse (class in here_location_services.responses), 85
 as_json_string() (here_location_services.responses.ApiResponse
 method), 85
 Auth (class in here_location_services.platform.auth), 91
 AuthenticationException, 72
 AutoCircleRegion (class in here_location_services.config.matrix_routing_config), 59
 AutoRequest() (here_location_services.ls.LS method), 75
 AutosuggestApi (class in here_location_services.autosuggest_api), 61
 AutosuggestResponse (class in here_location_services.responses), 87
 AVOID_FEATURES (in module here_location_services.config.matrix_routing_config), 60
 AVOID_FEATURESRegion (in module here_location_services.config.routing_config), 56
 AvoidBoundingBox (class in here_location_services.config.matrix_routing_config), 60
 AvoidBoundedRegion (class in here_location_services.config.matrix_routing_config), 58
 AvoidCircleRegion (class in here_location_services.config.matrix_routing_config), 58
 AvoidFeatures (class in here_location_services.config.routing_config), 56

B
 bicycle_route() (here_location_services.ls.LS
 method), 80
 BoundingBoxRegion (class in here_location_services.config.matrix_routing_config), 58
 browse_exception() (here_location_services.ls.LS method), 78
 BrowseResponse (class in here_location_services.responses), 86
 Bunch (class in here_location_services.config.base_config), 48

C
 calculate_isoline() (here_location_services.ls.LS
 method), 74
 can_route() (here_location_services.ls.LS method), 79
 CircleRegion (class in here_location_services.config.matrix_routing_config), 58
 ConfigException, 72

D

`DEST_WEATHER_PRODUCT` (in module `here_location_services.config.dest_weather_config`), 54

`DEST_WEATHER_UNITS` (in module `here_location_services.config.dest_weather_config`), 54

`DestinationWeatherApi` (class in `here_location_services.destination_weather_api`), 93

`DestinationWeatherResponse` (class in `here_location_services.responses`), 87

`DestWeatherProduct` (class in `here_location_services.config.dest_weather_config`), 53

`DestWeatherUnits` (class in `here_location_services.config.dest_weather_config`), 54

`discover()` (`here_location_services.ls.LS` method), 77

`DiscoverResponse` (class in `here_location_services.responses`), 86

`get_async_tour_planning_status()` (`here_location_services.tour_planning_api.TourPlanningApi` method), 100

`get_autosuggest()` (`here_location_services.autosuggest_api.AutosuggestApi` method), 63

`get_dest_weather()` (`here_location_services.destination_weather_api.DestinationWeatherApi` method), 93

`get_dest_weather()` (`here_location_services.ls.LS` method), 76

`get_geocoding()` (`here_location_services.geocoding_search_api.GeocodingSearchApi` method), 64

`get_isoline_routing()` (`here_location_services.isoline_routing_api.IsolineRoutingApi` method), 67

`get_reverse_geocoding()` (`here_location_services.geocoding_search_api.GeocodingSearchApi` method), 65

`get_search_browse()` (`here_location_services.geocoding_search_api.GeocodingSearchApi` method), 66

`get_search_discover()` (`here_location_services.geocoding_search_api.GeocodingSearchApi` method), 65

`get_search_lookup()` (`here_location_services.geocoding_search_api.GeocodingSearchApi` method), 66

`get_weather_alerts()` (`here_location_services.destination_weather_api.DestinationWeatherApi` method), 94

`get_weather_alerts()` (`here_location_services.ls.LS` method), 76

F

`Fleet` (class in `here_location_services.config.tour_planning_config`), 97

`from_credentials_file()` (`here_location_services.platform.credentials.PlatformCredentials` class method), 92

`from_default()` (`here_location_services.platform.credentials.PlatformCredentials` class method), 92

`from_env()` (`here_location_services.platform.credentials.PlatformCredentials` class method), 92

G

`generate_token()` (`here_location_services.platform.auth.Auth` method), 91

`geocode()` (`here_location_services.ls.LS` method), 73

`GeocoderResponse` (class in `here_location_services.responses`), 85

`GeocodingSearchApi` (class in `here_location_services.geocoding_search_api`), 64

`get()` (`here_location_services.apis.Api` method), 48

`get_apikey()` (in module `here_location_services.utils`), 89

`get_async_matrix_route_results()` (`here_location_services.matrix_routing_api.MatrixRoutingApi` method), 71

`get_async_matrix_route_status()` (`here_location_services.matrix_routing_api.MatrixRoutingApi` method), 71

`get_async_tour_planning_results()` (`here_location_services.tour_planning_api.TourPlanningApi` method), 101

`headers` (`here_location_services.platform.apis.Api` property), 90

`here_location_services` module, 47

`here_location_services.__version__` module, 47

`here_location_services.apis` module, 47

`here_location_services.autosuggest_api` module, 63

`here_location_services.config` module, 48

`here_location_services.config.autosuggest_config` module, 51

`here_location_services.config.base_config` module, 48

`here_location_services.config.dest_weather_config` module, 53

`here_location_services.config.isoline_routing_config` module, 52

`here_location_services.config.matrix_routing_config` module, 52

H

module, 58	IsolineRoutingAvoidFeatures (class in here_location_services.config.isoline_routing_config), 53
here_location_services.config.routing_config module, 55	IsolineRoutingTransportMode (class in here_location_services.config.isoline_routing_config), 52
here_location_services.config.search_config module, 60	
here_location_services.config.tour_planning_config module, 95	J
here_location_services.config.url_config module, 63	Job (class in here_location_services.config.tour_planning_config), 98
here_location_services.destination_weather_api module, 93	JobPlaces (class in here_location_services.config.tour_planning_config), 97
here_location_services.exceptions module, 72	L
here_location_services.geocoding_search_api module, 64	lookup() (here_location_services.ls.LS method), 78
here_location_services.isoline_routing_api module, 66	LookupResponse (class in here_location_services.responses), 86
here_location_services.ls module, 73	LS (class in here_location_services.ls), 73
here_location_services.matrix_routing_api module, 68	M
here_location_services.platform module, 89	matrix() (here_location_services.ls.LS method), 84
here_location_services.platform.apis module, 89	MATRIX_ATTRIBUTES (in module here_location_services.config.matrix_routing_config), 59
here_location_services.platform.apis.aaa_oauth2_api module, 89	matrix_route() (here_location_services.matrix_routing_api.MatrixRoutingApi method), 69
here_location_services.platform.apis.api module, 90	matrix_route_async() (here_location_services.matrix_routing_api.MatrixRoutingApi method), 70
here_location_services.platform.auth module, 91	MatrixAttributes (class in here_location_services.config.matrix_routing_config), 59
here_location_services.platform.credentials module, 92	MatrixRoutingApi (class in here_location_services.matrix_routing_api), 68
here_location_services.responses module, 85	MatrixRoutingResponse (class in here_location_services.responses), 86
here_location_services.routing_api module, 87	module
here_location_services.tour_planning_api module, 100	here_location_services, 47
here_location_services.utils module, 89	here_location_services.__version__, 47
I	here_location_services.apis, 47
ISOLINE_ROUTING_AVOID_FEATURES (in module here_location_services.config.isoline_routing_config), 53	here_location_services.autosuggest_api, 63
ISOLINE_ROUTING_TRANSPORT_MODE (in module here_location_services.config.isoline_routing_config), 52	here_location_services.config, 48
IsolineResponse (class in here_location_services.responses), 86	here_location_services.config.autosuggest_config, 51
IsolineRoutingApi (class in here_location_services.isoline_routing_api), 66	here_location_services.config.base_config, 48
	here_location_services.config.dest_weather_config, 53
	here_location_services.config.isoline_routing_config, 52
	here_location_services.config.matrix_routing_config, 58

here_location_services.config.routing_config, 55
 here_location_services.config.search_config, 60
 here_location_services.config.tour_planning_config, 95
 here_location_services.config.url_config, 63
 here_location_services.destination_weather_api, 93
 here_location_services.exceptions, 72
 here_location_services.geocoding_search_api, 64
 here_location_services.isoline_routing_api, 66
 here_location_services.ls, 73
 here_location_services.matrix_routing_api, 68
 here_location_services.platform, 89
 here_location_services.platform.apis, 89
 here_location_services.platform.apis.aaa_oauth2_api, 89
 here_location_services.platform.apis.api, 90
 here_location_services.platform.auth, 91
 here_location_services.platform.credentials, 92
 here_location_services.responses, 85
 here_location_services.routing_api, 100
 here_location_services.tour_planning_api, 100
 here_location_services.utils, 89

N

new() (here_location_services.responses.ApiResponse
 class method), 85

O

OPTIMISED_FOR (in module
 here_location_services.config.isoline_routing_config),
 53
 OptimisedFor (class in
 here_location_services.config.isoline_routing_config),
 53

P

patch_using_env() (here_location_services.platform.credentials.
 PlatformCredentials method), 93
 pedestrian_route() (here_location_services.ls.LS
 method), 83
 PlaceOptions (class in
 here_location_services.config.base_config), 50

PLACES_CATEGORIES (in module
 here_location_services.config.search_config),
 61
 PlacesCategories (class in
 here_location_services.config.search_config),
 60
 Plan (class in here_location_services.config.tour_planning_config),
 99
 PlatformCredentials (class in
 here_location_services.platform.credentials),
 92
 POLITICAL_VIEW (in module
 here_location_services.config.autosuggest_config),
 52
 PoliticalView (class in
 here_location_services.config.autosuggest_config),
 52
 PolygonRegion (class in
 here_location_services.config.matrix_routing_config),
 58
 post() (here_location_services.apis.Api method), 48
 post() (here_location_services.platform.apis.api.Api
 method), 90
 Profile (class in here_location_services.config.matrix_routing_config),
 59
 PROFILE (in module here_location_services.config.matrix_routing_config),
 59

R

raise_response_exception()
 (here_location_services.platform.apis.api.Api
 static method), 90
 RANGE_TYPE (in module
 here_location_services.config.isoline_routing_config),
 53
 RangeType (class in here_location_services.config.isoline_routing_config),
 53
 Relation (class in here_location_services.config.tour_planning_config),
 99
 request_scoped_access_token()
 (here_location_services.platform.apis.aaa_oauth2_api.AAAOAuth2Api
 method), 90
 reverse_geocode() (here_location_services.ls.LS
 method), 73
 ReverseGeocoderResponse (class in
 here_location_services.responses), 86
 route() (here_location_services.routing_api.RoutingApi
 method), 88
 ROUTE_MATCH_SIDE_OF_STREET (in module
 here_location_services.config.routing_config),
 56
 RouteCourse (class in
 here_location_services.config.routing_config),
 56

RouteMatchSideOfStreet	(class in here_location_services.config.routing_config), 56	(here_location_services.responses.MatrixRoutingResponse method), 86
ROUTING_MODE	(in here_location_services.config.base_config), 49	to_geojson() (here_location_services.responses.ApiResponse method), 85
ROUTING_RETURN	(in here_location_services.config.routing_config), 55	to_geojson() (here_location_services.responses.DestinationWeatherResponse method), 87
ROUTING_SPANS	(in here_location_services.config.routing_config), 55	to_geojson() (here_location_services.responses.IsolineResponse method), 86
ROUTING_TRANSPORT_MODE	(in here_location_services.config.routing_config), 56	to_geojson() (here_location_services.responses.MatrixRoutingResponse method), 86
ROUTING_TRANSPORT_MODE	(in here_location_services.config.routing_config), 56	to_geojson() (here_location_services.responses.RoutingResponse method), 86
ROUTING_TRANSPORT_MODE	(in here_location_services.config.routing_config), 56	to_geojson() (here_location_services.responses.TourPlanningResponse method), 87
RoutingApi	(class in here_location_services.routing_api), 87	to_geojson() (here_location_services.responses.WeatherAlertsResponse method), 87
RoutingMode	(class in here_location_services.config.base_config), 48	to_travel_times_matrix() (here_location_services.responses.MatrixRoutingResponse method), 86
RoutingResponse	(class in here_location_services.responses), 86	token (here_location_services.platform.auth.Auth property), 91
RoutingReturn	(class in here_location_services.config.routing_config), 55	token_still_valid() (here_location_services.platform.auth.Auth method), 91
RoutingSpans	(class in here_location_services.config.routing_config), 55	TooManyRequestsException, 72
RoutingTransportMode	(class in here_location_services.config.routing_config), 56	TOUR_PLANNING_AVOID_FEATURES (in here_location_services.config.tour_planning_config), 95
		TourPlanningApi (class in here_location_services.tour_planning_api), 100
S		TourPlanningAvoidFeatures (class in here_location_services.config.tour_planning_config), 95
Scooter	(class in here_location_services.config.routing_config), 56	TourPlanningResponse (class in here_location_services.responses), 87
scooter_route()	(here_location_services.ls.LS method), 82	Truck (class in here_location_services.config.base_config), 49
SearchCircle	(class in here_location_services.config.autosuggest_config), 51	truck_route() (here_location_services.ls.LS method), 81
SHIPPED_HAZARDOUS_GOODS	(in here_location_services.config.base_config), 49	
ShippedHazardousGoods	(class in here_location_services.config.base_config), 49	V
Show	(class in here_location_services.config.autosuggest_config), 52	VEHICLE_MODE (in here_location_services.config.tour_planning_config), 95
SHOW	(in module here_location_services.config.autosuggest_config), 52	VehicleMode (class in here_location_services.config.tour_planning_config), 95
solve_tour_planning()	(here_location_services.ls.LS method), 77	VehicleProfile (class in here_location_services.config.tour_planning_config), 96
solve_tour_planning()	(here_location_services.tour_planning_api.TourPlanningApi method), 100	VehicleType (class in here_location_services.config.tour_planning_config), 95
T		
to distnaces matrix()		

Via (class in *here_location_services.config.routing_config*),
57

W

WayPointOptions (class in
here_location_services.config.base_config), 51
WEATHER_SEVERITY (in module
here_location_services.config.dest_weather_config),
54
WEATHER_TYPE (in module
here_location_services.config.dest_weather_config),
54
WeatherAlertsResponse (class in
here_location_services.responses), 87
WeatherSeverity (class in
here_location_services.config.dest_weather_config),
54
WeatherType (class in
here_location_services.config.dest_weather_config),
54
WorldRegion (class in
here_location_services.config.matrix_routing_config),
59